

AD-A107 565

MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INF--ETC F/G 9/2  
COMPUTER INTERPRETATION OF A DYNAMIC IMAGE FROM A MOVING VEHICLE--ETC(U)  
MAY 81 T D WILLIAMS  
COINS-TR-81-22

N00014-75-C-0459

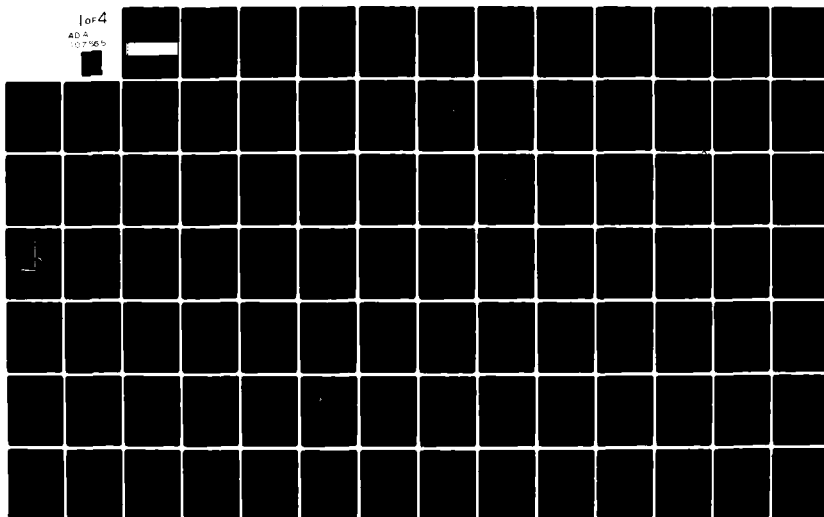
NL

UNCLASSIFIED

for 4

ADA

107565

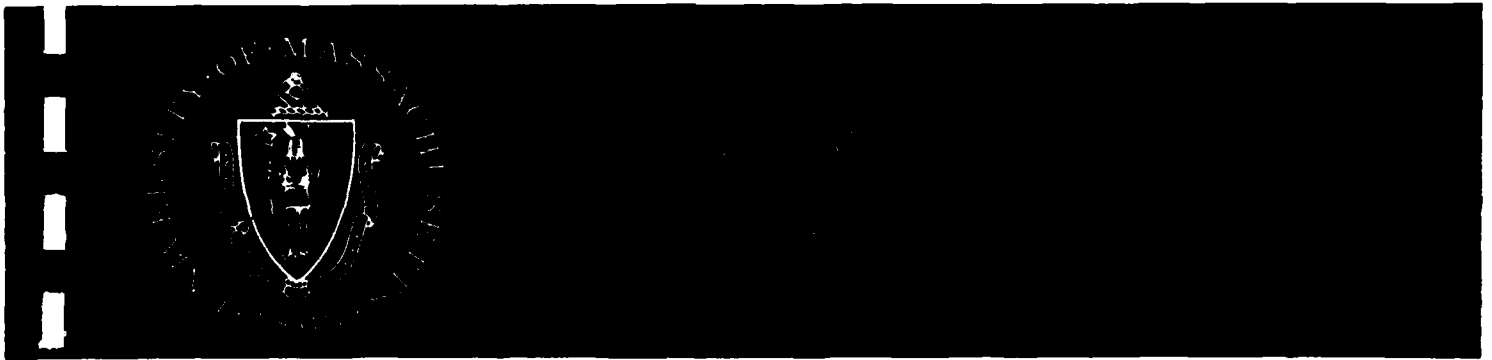


AD A 110 068

LEVEL

12

# Computer and Information Science



This document has been approved  
for public release and its  
distribution is unlimited.

01 11 00 068

**LEVEL**

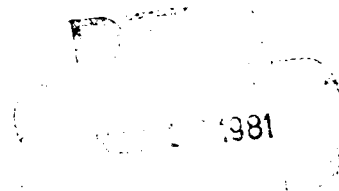
12

COMPUTER INTERPRETATION OF A DYNAMIC  
IMAGE FROM A MOVING VEHICLE

Thomas D. Williams

COINS Technical Report 81-22

May 1981



This work was supported by the Office of Naval Research under grant number N00014-75-C-0459 and the National Science Foundation under grant number DCR75-16098.

This document has been approved  
for public release and its  
distribution is unlimited.

COMPUTER INTERPRETATION OF A DYNAMIC  
IMAGE FROM A MOVING VEHICLE

A Dissertation Presented

By

THOMAS DELL WILLIAMS

Submitted to the Graduate School of the  
University of Massachusetts in partial fulfillment  
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May

1981

Department of Computer and Information Science

Accession For	
HTIS GRAM	<input checked="" type="checkbox"/>
DEIS TIS	<input type="checkbox"/>
Enrollment	<input type="checkbox"/>
Justification	<input type="checkbox"/>
By	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A	



Thomas Dell Williams

1981

All Rights Reserved

This work was supported by  
The Office of Naval Research  
grant number N00014-75-C-0459  
and  
The National Science Foundation  
grant number DCR75-16098

COMPUTER INTERPRETATION OF A DYNAMIC  
IMAGE FROM A MOVING VEHICLE

A Dissertation Presented

By

Thomas D. Williams

Approved as to style and content by:

Edward Riseman  
Edward Riseman, Chairperson of Committee

Alan Hanson  
Alan Hanson, Member

Charles Clifton  
Charles Clifton, Member

William Kilmer  
William Kilmer, Member

Edward Riseman  
Edward Riseman, Department Head  
Computer and Information Science

## DEDICATION

This work is dedicated to the memory of Cesare Parma, a fine individual and a thorough scientific investigator. His part in the VISIONS team will always be remembered.

## ACKNOWLEDGMENT

The untiring efforts and sense of direction provided by Professors Edward Riseman and Allen Hanson were instrumental in the production of this dissertation. I would also like to thank Professors William Kilmer, Charles Clifton and Michael Arbib for their assistance. The VISIONS group at the University of Massachusetts is too large to name, but special thanks to Frank Glazer, Ralph Kohler, Daryl Lawton, John Lowrance, Paul Nagin, and John Prager. The Office of Naval Research and the National Science Foundation supported the research that this dissertation reports.



ABSTRACT

Computer Interpretation of a Dynamic  
Image from a Moving Vehicle

May 1981

Thomas Dell Williams

A.S. Northern Essex Community College

B.S. University of Massachusetts

M.S. University of Massachusetts

Ph.D. University of Massachusetts

Directed by: Professor Edward M. Riseman

The goal of this thesis is the design and implementation of a computer program that constructs an interpretation of images of a natural scene, in particular one imaged while the camera is in a moving automobile. The succession of images is to be interpreted in terms of surfaces and objects in three-dimensional space.

The agreement between image dynamics and an internal surface model of the environment is measured by comparing a pair of temporally disparate images (two movie frames). Using the model, an image taken at one location can be transformed into a synthetic image of the scene as it would be viewed from another location. This synthesis accounts for point displacements and

occlusion effects as predicted by the internal model. Differences between the real and the synthetic images are then used as an error measure in a search that refines the model. Once the model is refined, unresolved errors are used to correct the initial surface model by resegmenting the image into a better approximation of the surfaces in the environment.

This surface model refinement is followed by an object identification phase. Size and color attributes measured from the derived internal model are compared with stored attributes for objects. The result is the identification of some of the scene objects.

## TABLE OF CONTENTS

ACKNOWLEDGEMENT	v
LIST OF TABLES	xi
LIST OF ILLUSTRATIONS	xii
Chapter	
I. INTRODUCTION	1
I.1 Goals	3
I.2 Image Interpretation via Motion	4
I.3 Depth from Motion	7
I.4 Surface Interpretation	8
I.5 Object Interpretation	12
Chapter	
II. REVIEW OF SCENE ANALYSIS	14
II.1 The Elements of Scene Analysis	14
II.1.1 Sensor	15
II.1.2 Image	16
II.1.3 Features	17
II.1.4 Aggregations of Features	21
II.1.5 Interpretation	23
II.2 Problems in Static Scene Analysis	24
II.3 Static Scene Analysis Systems	29
II.3.1 Theoretical Visual Systems	29
II.3.2 General vs. Specialized Systems	33
II.3.3 Blocks World Scenes	34
II.3.4 Real World Scenes	37
II.3.5 Query-directed Systems	40
II.3.6 General-purpose Systems	44
II.3.7 Static Scene Analysis Summary	50
II.4 Moving Image Analysis Systems	51
II.4.1 Vector Field Techniques	53
II.4.2 Tracking Techniques	62
II.4.3 Predictive Modeling	68
II.4.4 Relaxation Techniques	69
II.4.5 Moving Image Analysis Summary	72
Chapter	
III. THE SURFACE INTERPRETATION	79
III.1 Representation and Issues	82
III.1.1 Problems of Startup and Continuation	82
III.1.2 Moving Image Projection	84
III.1.3 Focus of Expansion	92

	III.1.4 Assumption of Non-rotating Camera	95
	III.1.5 Surface Orientation	100
	III.1.6 Features Chosen	104
	III.1.7 The Initial Model	109
III.2	The Surface Interpretation Process	112
	III.2.1 Image Synthesis	118
	III.2.1.1 Occlusion	119
	III.2.1.2 Interpolation	128
	III.2.2 Search Using Error Images	132
	III.2.3 Search for FOE	136
	III.2.4 Search for the Z and Y Values for Surfaces	142
	III.2.5 Decoupling the Z and FOE	145
	III.2.6 Resegmentation	153
	III.2.7 Surface Merging	158
	III.2.8 Summary	158
Chapter		
IV.	DATA AND RESULTS FOR SURFACE REPRESENTATION	160
IV.1	Data	161
	IV.1.1 Collection and Registration	161
	IV.1.2 Scene Measurements	172
IV.2	Segmentation	175
	IV.2.1 The Subimage	177
	IV.2.2 The Averaged Image	182
IV.3	Experiments	185
	IV.3.1 Experiment #1	187
	IV.3.2 Experiment #2	194
	IV.3.3 Experiment #3	198
	IV.3.4 Experiment #4	203
	IV.3.5 Experiment #6	207
	IV.3.6 Experiment Summary	216
Chapter		
V.	OBJECT INTERPRETATION	219
V.1	Representation	221
	V.1.1 Levels of Abstraction	222
	V.1.2 Short and Long Term Knowledge	225
V.2	Knowledge Sources	227
	V.2.1 Data Collection	231
	V.2.2 Prototype Formation	233
	V.2.3 Matching	237
	V.2.4 Size Prototype and Matching	241
V.3	Results	244
	V.3.1 Results for the Color KS	244
	V.3.2 Results for Size Matching	252
	V.3.3 Combination of Spectral Attribute and Size Results	252

Chapter		
VI.	CONCLUSIONS	258
VI.1	Summary	258
VI.2	Sources of Error	261
	VI.2.1 Bad Segmentation	261
	VI.2.2 Blur	262
	VI.2.3 Resolution	263
	VI.2.4 Surface Orientation	264
VI.3	Suggested Future Improvements	264
	VI.3.1 Better Data	264
	VI.3.2 Third Orthogonal Plane	265
	VI.3.3 Automatic Foveation	265
	VI.3.4 Implementation of Dynamic Object Representation	266
	VI.3.5 More Objects	266
	BIBLIOGRAPHY	267
	APPENDIX I	274
	APPENDIX II	276
	APPENDIX III	278

# LIST OF TABLES

1	Registration	167
2	FOE Search	197
3	FOE Search Unweighted	206
4	FOE Weighted Search	213
5	Results for Color KS	245
6	Results for Color KS	250
7	Results for Size KS	254
8	Results for Combining KSs	256

## LIST OF ILLUSTRATIONS

Figure		Page
1	Scenario of Thesis	5
2	System Design	6
3	Motion Systems	9
4	Images	18
5	Edge Feature	20
6	House Segmentation	22
7	Image Intensity Differences	26
8	Robert's System	36
9	Interpretation Gioded Segmentation	38
10	Garvey's System	41
11	Bullock's System	46
12	The VISIONS System	48
13	Correspondence Problem	52
14	Vector Fields	54
15	Nevatia's Problem	59
16	Thompson's Estimation	60
17	Nagel's Technique	63
18	Price's System	66
19	Martin and Aggarwal's System	70
20	Interpixel Feature Problem	74
21	Movement of a Vertex	76
22	Overall Diagram	81
23	Sequence of Frame Pairs	83
24	Coordinate System	85
25	Three Film Angles	87
26	Skew Coordinates	88
27	Relations on Skewed System	93
28	Real and Ideal Images	97
29	Examples of Orientation	102
30	Horizontal Surface	105
31	Feature Derivation	107
32	Initial Segmentation Process	111
33	FOE and Displacement Vectors	114
34	Effect of Moving FOE	115
35	Changing Z	116
36	3 Phases of Interpretation	117
37	Interpolation	120
38	Occlusion Geometry	122
39	Motion Occlusion	123
40	Projecting Surface Model	125
41	Projection Flow Chart	127
42	Interpolation Flow Chart	129
43	Interpolation	131
44	Search Process	134

45	Search For Lowest Error	137
46	Nine Focci	138
47	FOE Search Pattern	140
48	Incorrect Choice of FOE	147
49	Displacements for FOE	150
50	Weighting Function	152
51	Resegmentation	156
52	Pan and Tilt	162
53	Camera Pointing	163
54	Position Error	165
55	Residual Error	168
56	Data	169
57	Image 45	171
58	Histograms	173
59	Four Features	178
60	Subimage 45	179
61	Segmentation for 45	181
62	Subimage for 51	183
63	Segmentation for 51	184
64	Segmentation for pair #4	186
65	Experimental Results	189
66	Surface Model	190
67	Error Functions	192
68	Error <u>vs</u> Iteration	193
69	FOE Search	196
70	Error Images	200
71	Resegmented Model	201
72	Unweighted FOE Search	204
73	Samples from the Z Search	212
74	Weighted FOE Search	214
75	Number of Surfaces Changing	215
76	Image and Resulting Model	217
77	Abstraction Levels	223
78	Example Interpretation	226
79	An Instantiated Object	229
80	Data Base Make-Up	232
81	Average and S.D. Features	235
82	Matching Function	239
83	Size Functions	243
84	Confidence Value Distributions	247
85	Segmentation of Averaged Image	249
86	Object Interpretation	257



## C H A P T E R I

### INTRODUCTION

Humans rely heavily on vision to recognize, measure and appreciate their environment. The speed, accuracy, and reliability of human vision challenges those who would construct an artificial system with similar performance. No one has yet succeeded in constructing such a system, although many advances in the field of scene analysis have taken us closer to that goal.

Theorists have proposed (Gibson 1950, Marr 1977) that sufficient information exists in static and dynamic images to derive an understanding of the physical environment depicted in the images. These theories suggest that object/background separation, the size and shape of objects, and distances to them can be determined without prior knowledge of the specific objects that appear.

Pictorial cues such as texture, shadows, and occlusion that are available in static monocular images can be used to infer important depth information. Humans can understand a static image as surfaces and objects in the physical world. This implies that there is sufficient information preserved in a static image to

allow the reconstruction of a plausible three dimensional scene.

Static scene analysis systems (Barrow 1978, Hanson 1978b, Bullock 1978) exploit pictorial depth cues in an attempt to derive an interpretation of an image as surfaces and objects. Once depth information is obtained, the orientation of surfaces, the identity of objects, and the spatial layout of the scene can be determined. However, the problem of automatic and reliable inference of depth from static cues remains unsolved for general static scenes, although considerable progress is being made at understanding what the cues are, and how to use them.

The motion of imaged scene points from a moving camera provides direct rather than inferred depth information. The convincing fidelity of depth conveyed in motion pictures demonstrates the ability of a dynamic image to preserve depth information in a direct and accurate manner.

### I.1 Goals

The goal of this thesis is the design and implementation of a computer program that constructs an interpretation from images of a natural scene, in particular one imaged while the camera is in a moving automobile. Motion cues derived from successive frames of a movie will be exploited to allow the moving image to be interpreted in terms of surfaces and objects in three-dimensional space.

A second and related goal is the identification of objects based on color, texture and size. This goal is an exercise in the structuring of high level knowledge about scenes and objects. An object identification system is presented which uses as input the results of the surface interpretation process.

The goals are met through a set of experiments that are presented in chapters IV and V. Our methodology for system development involved the testing of each subsystem individually by providing it with the information that other subsystems would produce in the completed system. Then the entire system's behavior was tested.

## I.2 Image Interpretation via Motion

In this thesis techniques from the field of static scene analysis are extended by incorporating depth cues derived from a dynamic image. We chose to analyze dynamic images because important depth information is available for use in directly segmenting objects from their background. We propose a group of processes for interpretation of a natural scene that is successively photographed by a camera in motion (Figure 1). These movie frames are to be interpreted in terms of surfaces and objects of the scene. Motion information leads to a description of the scene in terms of surfaces, which in turn leads to a description of the scene in terms of object identities (see figure 2).

One subsystem is explored in great detail since it is the basis for an approximate surface interpretation of the physical environment. A model is hypothesized depicting the three-dimensional positions of scene surfaces relative to the camera. Then, the motions of image features are used to refine this model. The surface interpretation provides both the object/background segmentation and size measurements for object identification. An object interpretation is

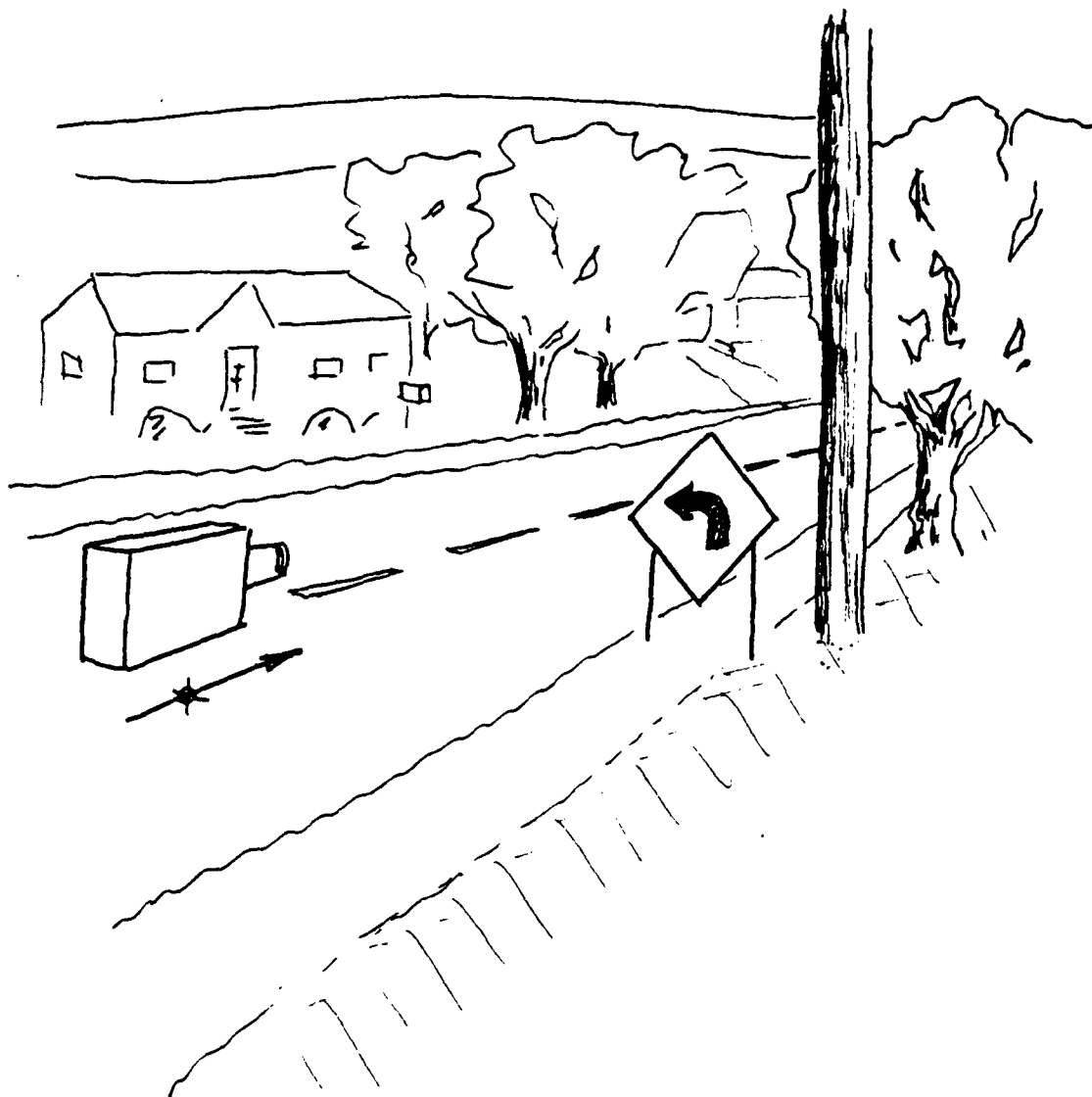


Figure 1 The scenerio that this thesis exams is  
that of a camera in motion.

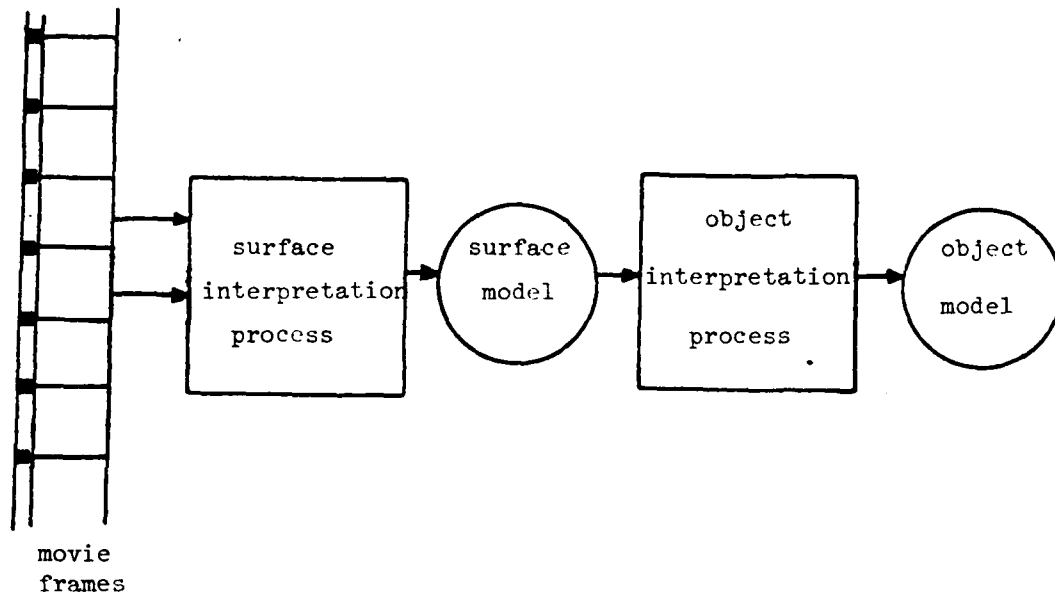


Figure 2 Overall system design

produced by comparing both spectral features from the image and size information from the surface interpretation to prototypes which are associated with stored object names.

### I.3 Depth from Motion

To determine the distance from the camera to scene points based on the positions of points on the image plane, images of the scene taken in two physically disparate locations are used. For each scene point that appears in both images, an inter-image displacement can be measured. By simple triangulation the three-dimensional position of each point with respect to the camera can be determined. This technique is called "motion stereo" because it involves the comparison of a pair of images taken from a moving camera.

Automatic discovery of the displacement of projections of a scene point between two images is called the "correspondence problem" (Ullman 1978, Quam 1974a), or "stimulus organization problem" (Burt 1976). The thrust of most research in stereo image understanding has been the reliable and fast computation of inter-image correspondence of points. The system presented here develops an interpretation of distances using an

hypothesize-test strategy. An hypothesized three-dimensional interpretation - expressed as a model of scene surfaces - predicts image dynamics which are tested through inter-image comparisons. This is in contrast to the more common motion detection techniques (Quam 1974, Prager 1979, Thompson 1979) that detect image dynamics to generate a three-dimensional model of the scene without any prior hypotheses about the surfaces in the scene (figure 3).

#### I.4 Surface Interpretation

Surfaces are the boundaries of objects, and are the places where the scene illumination is reflected. In general surfaces are curved, and very sharp curvature of surfaces, such as where two faces of a cube meet, are called surface edges or discontinuities. Curved three-dimensional bodies can be modeled as composites of surface patches, joined at space curves (York 1979).

The representation utilized here involves planar surfaces at orthogonal orientations within the viewing geometry. Although this representation does not accurately reflect the nature of real surfaces, our premise is that it is sufficient for recognizing objects at a distance. The size dimensions of height, width, and



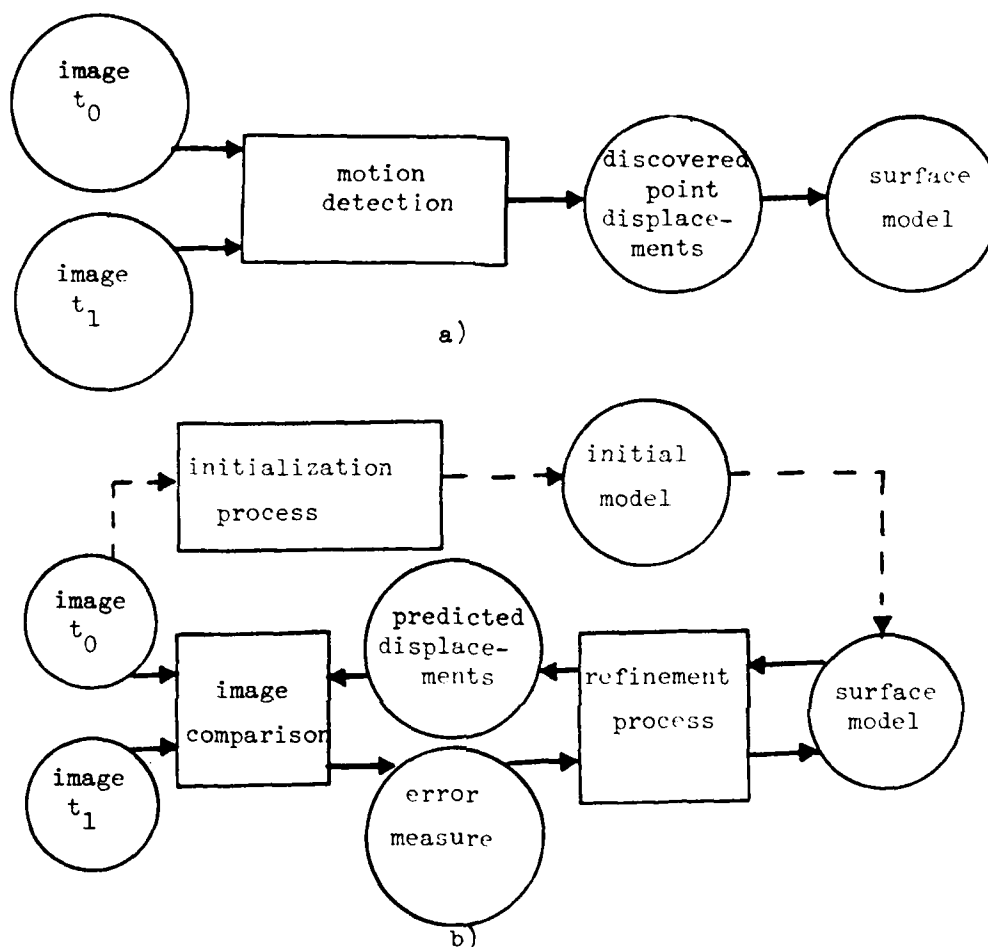


Figure 3 a) Typical motion detection systems analyse data to discover a model, b) the system presented here hypothesizes a surface model and then refines it.

depth of a telephone pole, or a tree can be approximated reasonably well if the objects are represented as rectangular solids.

The surface interpretation process involves the combination of static image analysis and the analysis of image dynamics. A surface interpretation is first hypothesized on the basis of a coarse static analysis. Then, this interpretation is used as an initial scene model to predict image dynamics. The model is refined on the basis of the agreement between image dynamics and the model. An image taken at one location is transformed into a synthetic image of the scene as it would be viewed from another location, given the scene model. This synthesis accounts for point displacements and occlusion effects as predicted by the surface model. Differences between the real and the predicted images are then used as an error measure in a search that refines the surface model (see figure 3).

Once the model is refined, unresolved difference values are detected wherever two-dimensional image dynamics disagree with the three-dimensional surface model. Thus, errors in the initial segmentation of the image into surfaces would be detected by these difference

values. The initial surface model can be corrected by resegmenting the image into new surfaces which better account for the image dynamics.

This hypothesize-test process could be used to refine the orientation, distance and curvature of each surface which is hypothesized. Such a process would be computationally expensive and its results too detailed for many purposes. We simplify by assuming that all surfaces are planar and oriented in either of two directions.

The two-orientation representation is not an inherent system limitation, but rather an efficient means for deriving a surface model that has sufficient detail to interpret the positions and identities of objects in our test scenes. The two orientations chosen are parallel to the ground plane, which we call "horizontal", and parallel to the image plane, which we call "vertical". The road, grass, and soil are all oriented in the ground plane. All other objects can be approximated by flat surfaces parallel to the image plane, for in our scene no objects (other than the horizontal ones) have large depth disparities across them. For images with a long planar surface, such as the

wall of a building parallel to the direction of travel, this two-orientation representation would need to be extended to three orientations.

### 1.5 Object Interpretation

The term "object" carries many meanings. We refer to the items in the scene that are physically separate entities as objects. Trees, telephone poles, signs, people, the sky, and the road are all examples of objects. Except for the sky, the objects that we deal with are solid, and usually touch one another for support. Transparent objects do not appear in the images and there is no mechanism for dealing with the appearance of several objects in the same image location.

Within our definitions, the interpretation of natural outdoor images culminates in an understanding of the spatial layout and identity of objects in the scene. To determine identity, systems must match the size and color of objects detected in the image to the size and color associated with the stored concepts of known objects.

In static scene analysis systems, various techniques are applied, each with its assumptions about the scene, to obtain object size from static image features. With the assumption of an accurately modeled ground plane and an assumption of the orientation of a given surface, the size and distance to that surface can be determined directly. Alternatively, if there is prior knowledge of the expected position or size of the objects appearing it is possible to use matching techniques to directly obtain spatial relationships.

We chose to avoid implementing a system requiring the extensive use of knowledge which is specific to the particular scene. Thus, our approach is to use information from the image, in a bottom-up fashion, to produce a description of the scene in terms of surfaces. The surface description is then used, again bottom-up, to derive an object interpretation.

## C H A P T E R   I I

### REVIEW OF SCENE ANALYSIS

This chapter reviews the field of scene analysis. First, some concepts are described which are used in the analysis of images. Then, the problems involved in computer interpretation of images are briefly examined. This is followed with a review of the literature pertaining to the analysis of static images. The final section deals with the issues and the literature pertaining to the analysis of dynamic images. The reader is directed to section II.4 if he is knowledgeable of the static scene analysis problems and literature.

#### II.1 The Elements of Scene Analysis

Scene analysis is a field of study aimed at automatic interpretation of images of scenes. Although the techniques are as varied as the domains of application, there are certain elements common to all scene analysis systems. These elements are 1) a scene, 2) a sensor, 3) an image, 4) extracted features, 5) aggregations of features often called image segmentations, and 6) an interpretation. In each scene analysis system these elements and the interactions

between them are tailored to meet specific goals. We begin by describing the elements, and follow by examining the way they interact in several systems.

II.1.1 Sensor. The device that records information from a scene is called a sensor. Sensors are transducers that convert scene illumination into another form of energy (usually electrical) that can be measured. A camera is employed to image (form a projection of) the light flux present at some point in a scene. The transduction takes place in the photographic emulsion if a film camera is used, or in the photo-sensitive target if a television camera is used. The corresponding subsystem in the human visual system is the eye where the light flux that impinges on the retina causes electro-chemical activity.

Some scene analysis systems are designed to integrate active as well as passive sensor information. Active sensors are coupled with their own illuminators so that the nature of the signal being sensed is known, while passive sensors record illumination that exists naturally in the environment. The use of radar range-finding in the application of military target interpretation, laser range-finders for scene analysis (Duda 1979), and the use of ultrasound imaging systems

for non-invasive medical applications are all examples where surface distance information is available directly from an active sensor. Our goal is to derive surface information from a passive sensor - a moving camera - that records light flux as it occurs in a natural daytime environment.

II.1.2 Image. An image is a projection of reflected scene illumination onto a surface. We deal with existing light flux and flat image planes in our system. Systems have been designed that make use of other illuminants such as radar waves and infrared light, and other projection surfaces, such as spherical (Badler 1976). We record our images and usually refer to an image as an entity that exists now, although it was recorded in the past. A moving image is recorded as temporally disparate frames, each considered a separate image with an associated time index.

To facilitate computer processing, images are quantized in a regular array. Various patterns, such as hexagonal and rectangular arrays have been used, but for our work the more common square array has been adopted. Each unit square is an element of an array and is given a value that is the average of sensor values within the



boundaries of the element. These picture elements are called pixels, and are often referred to as image points.

Temporal quantization occurs in a moving image recording system. We use movie frames that are recorded at 18 frames per second, and select for analysis frames that are nine apart in the sequence, resulting in an effective frame rate of two per second (see figure 4).

II.1.3 Features. A feature is some abstraction of image information indicating points or areas of significance. Features vary in complexity and usefulness (Bullock 1974) and, therefore, are selected for each domain of application in scene analysis. The majority of scene analysis features fall into the categories of "point", "edge", or "texture".

Point features are those which can be derived from the information available at one pixel. In monochromatic images the only point feature is intensity. In multi-spectral images, various color features can be computed from the red, green, and blue values that are typically recorded at each pixel.

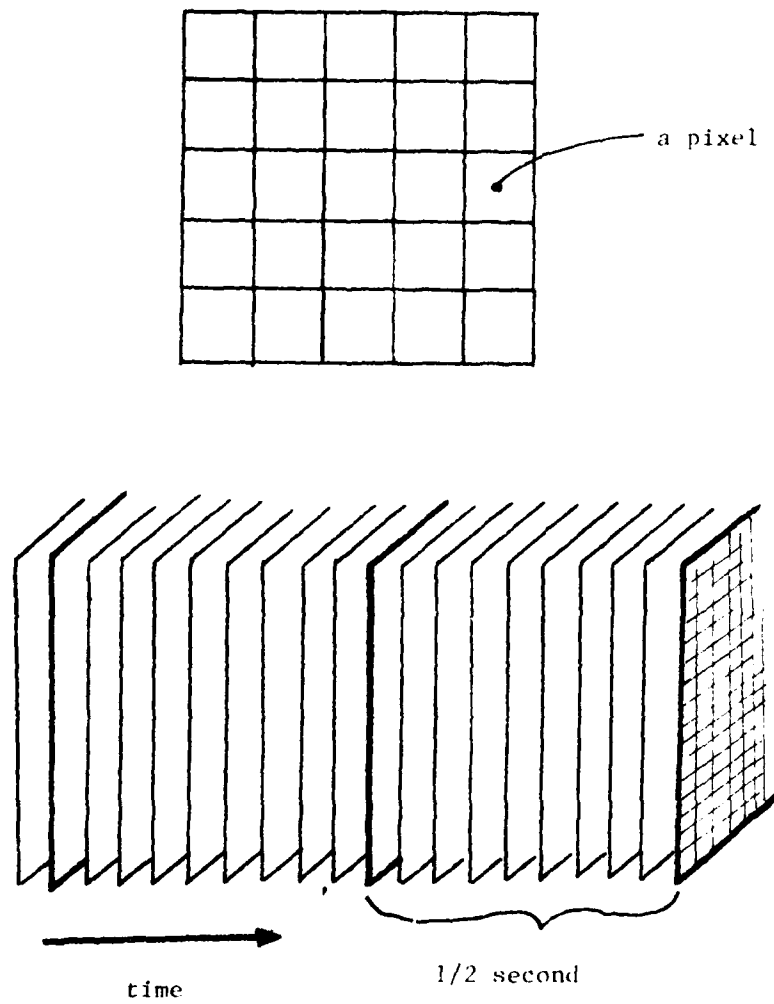


Figure 4. Images for static and dynamic scene analysis

Edge features are the result of inter-pixel differencing. These features are consistent with the observation that important scene information can be conveyed by simple line drawings (Attneave 1954). Typically, the average of point values from two adjacent areas are compared (see figure 5). We call the difference between these two averages the "response" of the edge operator. When the difference exceeds some threshold, an edge is placed between the two areas. Alternatively, a confidence of "edgeness" can be assigned to a short line segment placed between the two areas.

Texture features are typically either statistical or geometrical abstractions of areas of the intensity image. We are only concerned with the use of texture for the recognition of objects. The use of simplified texture features for this purpose is presented in chapter V.

Point features, such as color and intensity, are alone insufficient to provide an interpretation of a scene. According to one line of reasoning (Bullock 1977), point features are highly variant with respect to object models because of lighting conditions, whereas the shape of connected edges is a more useful and invariant feature for eventual interpretation. We agree that

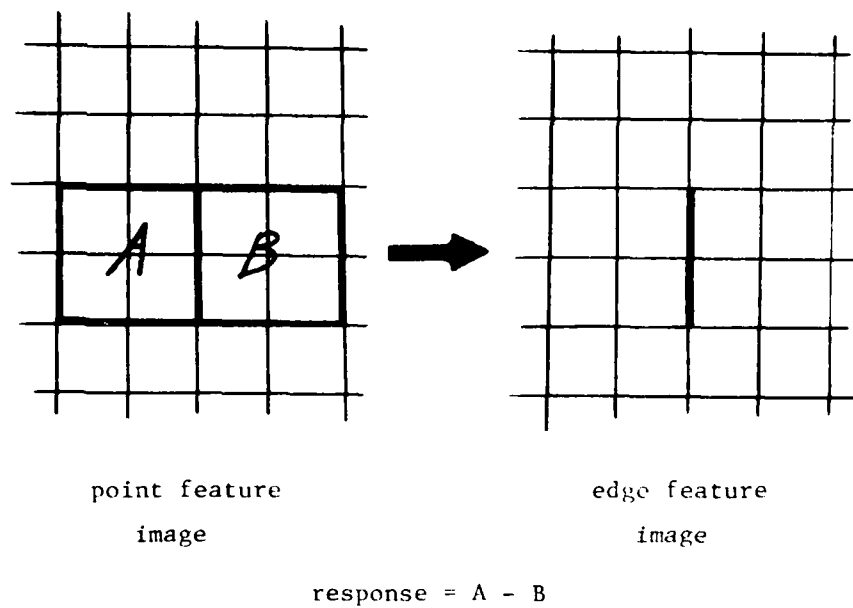


Figure 5. A simple edge feature might be the difference two adjacent 2 x 2 areas.

features that are used for interpretation must be invariant, but only with respect to the particular interpretations performed. Problems with the use of edge features for motion analysis will be addressed at the end of this chapter.

II.1.4 Aggregations of features. Features are usually aggregated into abstractions that serve to segment the image into meaningful areas. In scene analysis these aggregations involve regions, corresponding to areas of points which are similar in some feature; line segments, corresponding to boundaries between areas that are dissimilar with respect to some feature(s); and vertices, corresponding to the junction of line segments (Hanson 1976). The result of aggregation is an image which is partitioned into regions. We call a partitioned image a "segmentation" (see figure 6).

This intermediate-level image description is intended to provide an interpretation subsystem with information that can be readily compared with object models. Our system relies on regions (the aggregated point feature), to define localities in which interpretation processes act.

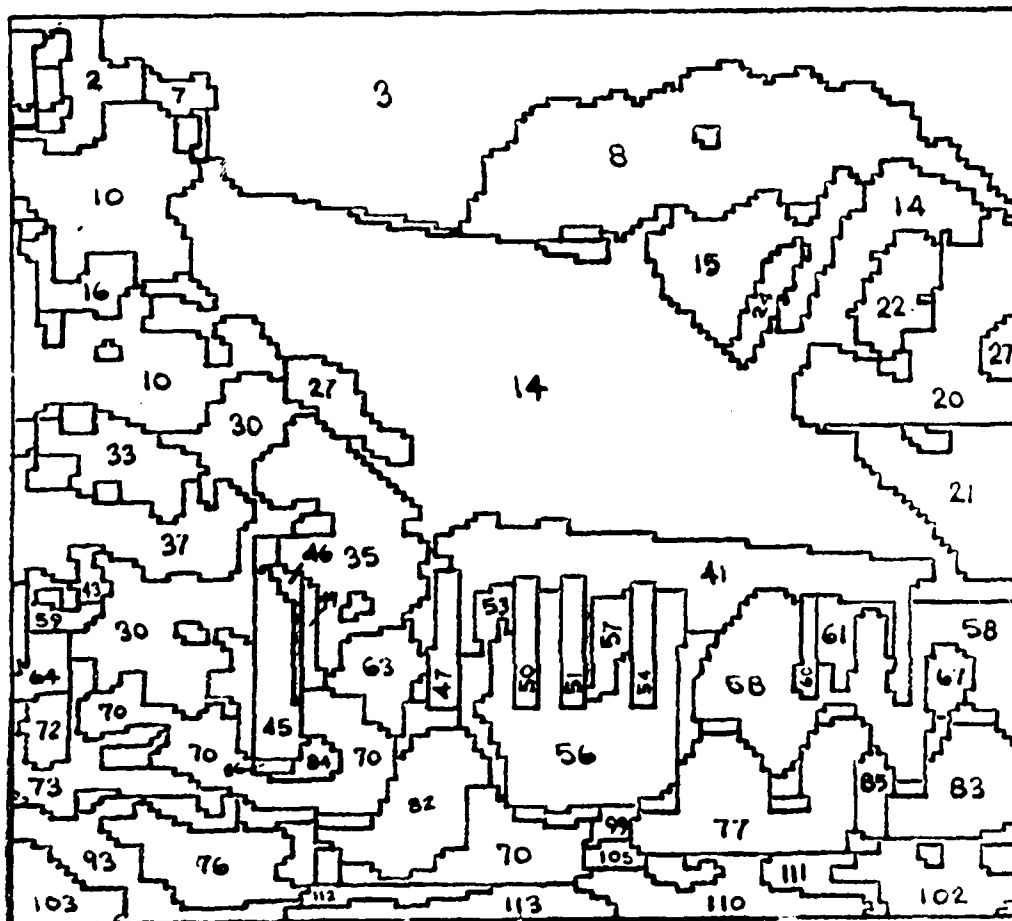


Figure 6. A segmentation of a house scene with the regions labeled by number

II.1.5 Interpretation. Interpretation refers to the derivation of a description of the scene in terms of the identity and position of objects that appear in an image. Scene analysis systems vary considerably as to what their interpretations will consist of, and in what domains they are applicable. Some schemes are designed to examine a novel approach to the application of knowledge in an artificial intelligence problem (Freuder 1973), while others are intended to understand arbitrary outdoor scenes containing any of numerous possible objects (Hanson 1978, Ohlander 1975).

Interpretation processes use object models, aggregated features and a matching strategy to derive identities and spatial relationships of scene objects. Various matching systems and strategies have been explored, and are briefly examined below (section II.3).

Object identity can be derived by matching color and texture features of regions with models of objects stored in terms of those features. This technique is applicable for objects that are relatively invariant in color and texture, such as the sky, trees and grass. Other features, such as line segments and vertices (Roberts 1965, Waltz 1972) or shape (York 1980), are used in the

matching process when objects can be identified by these characteristics.

The identification of objects is often enhanced by first interpreting the scene in terms of the disposition of scene surfaces. Three-dimensional characteristics of objects can then be used in the matching process (Marr 1977 ). A model of the scene in terms of surfaces seems a natural intermediate interpretation that bridges the gap between image features and object identification. This intermediate description is called a surface interpretation.

## II.2 Problems in Static Scene Analysis

Obviously, the extraction of image areas that correspond to separate objects can be based on differences in depth (Duda 1979). Unfortunately, no feature of a static image is guaranteed to indicate discontinuities in depth. Point and edge features can indicate discontinuities in intensity and color that arise from several phenomena which are unrelated to differences in depth.



Scene analysis systems are effective because discontinuities in depth are very often associated with other scene differences that give rise to intensity differences (Gibson 1950). Static scene analysis, until recently, had ignored this fact and still achieved acceptable levels of performance (Hanson 1978b, Bullock 1978, Levine 1978). The reader is referred to (Barrow 1978) and (Horn 1970, Horn 1977) for a more comprehensive description of the relationship between scene characteristics and image intensities than that presented below.

The intensity values recorded from an image are the result of three scene-dependent factors. They are 1) the magnitude of illumination falling on the imaged scene, 2) the type of material composing each surface of the scene, and 3) the orientation of each surface relative to the viewer and light sources. We refer to these three characteristics of the scene as the illumination, reflectance, and orientation respectively. See figure 7 for examples of these differences.

The illumination in a natural outdoor scene comes from three sources. They are the direct rays from the sun, the omnidirectional (diffuse) light from the sky,

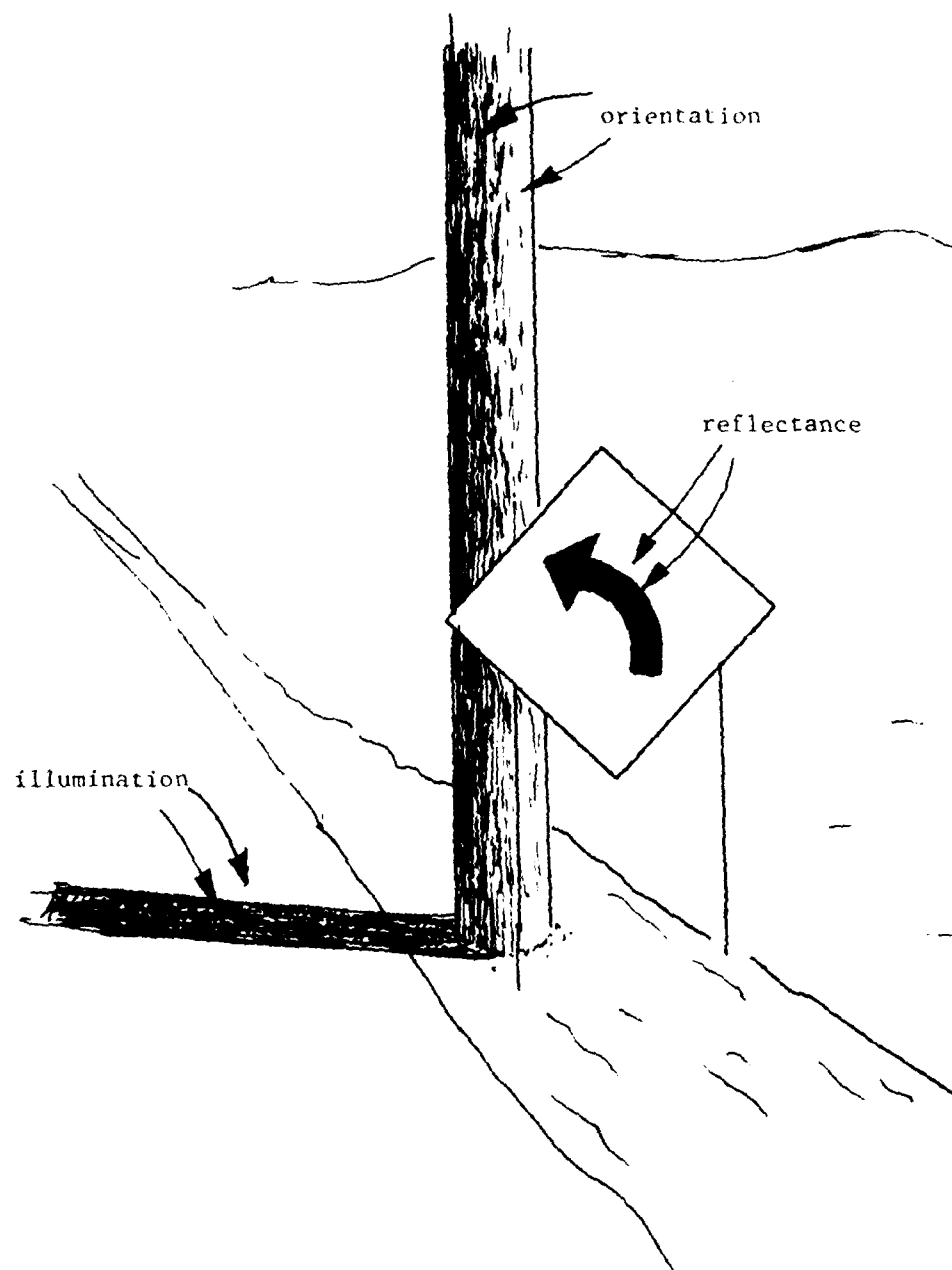


Figure 7. Examples of differences in image intensity due to differences in the scene.

and reflected light from other objects.

Differences in the amount of illumination are mostly accounted for by shadowing of direct sunlight. Shadowing of sky light and reflected illumination does not play as important a role as illumination differences in our scenes, although in some situations these effects are visible.

Differences in reflectance are due to the composition of the material where the light impinges. Surface reflectance characteristics are determined by the molecular structure near the surface of the object. This property provides the rich variety of coloration in images of natural scenes. It is responsible for the greenness of grass and leaves, for the contrast between the background of a sign and its message, and for the clarity of a dash painted on a road. Static scene analysis systems derive their interpretations primarily (although not entirely) through differences in reflectance.

Differences in the orientation of a surface with respect to the camera and illuminating source create differences in imaged intensity (Horn 1970). The formulation of the percentage of the light that is

reflected depends on the reflectance characteristic. Metallic surfaces tend to reflect most of the impinging light in one direction while matte surfaces reflect light evenly in all directions. For any surface of constant reflectance the image will be of constant intensity where the orientation is constant. If a surface is curved its orientation varies. For most surface reflectance characteristics, a variation in orientation will produce a variation in imaged intensity values.

Orientation differences are responsible for some of the fine textures of trees and grass because the leaves are at a variety of orientations. This characteristic also accounts for the intensity gradient across cylinders (such as telephone poles) and for the highlights imaged from metallic and glossy painted surfaces as found on automobiles.

Segmentation algorithms are generally capable of detecting all intensity and color differences in images, but are not capable of distinguishing one source of difference from another. A possible exception is the work being done by Barrow and Tenenbaum (Barrow 1978) where, in restricted domains, it appears possible to determine depth discontinuity.

Some researchers have found it beneficial to err on the side of excessive segmentation and allow some interpretive process to decide what regions must be merged, or what line segments must be ignored, in order to compose a depth segmentation. Our system provides an answer to this scene analysis problem by merging and splitting regions based on their behavior over time.

### II.3 Static Scene Analysis Systems

In this section, we discuss a few static monocular systems which are primarily aimed at interpreting natural scenes. Several theoretical scene analysis systems, and several that have been implemented are discussed. The systems are presented in categories, and are followed by a summary.

II.3.1 Theoretical visual systems. Marr proposes that there are three stages of visual processing, that each stage has a related representational structure, and that these structures must be understood before algorithms should be implemented to perform general visual tasks (Marr 1977). Once these representations are understood, the computational problems (hardware and software) can be devised and a resulting general purpose visual system can be realized.

The application of top-down design by Marr coincides with hierarchical decomposition (Simon 1969). Marr points out that the structure of a representation is determined both by the form of the information given to each process and the form that each process is expected to produce. Thus, at each stage of problem decomposition the interaction between modules of processing activity must be expressed in some representation.

Marr shows that the approach taken in some scene analysis systems to segment an image is misguided. In order to segment an image into objects, a system must employ knowledge about the particular scene. No clear way exists to choose what knowledge should be applied to produce segmentations.

The proposed solution is a three-stage system where 1) intensity and geometry of the image are used to produce a "primal sketch", 2) the sketch is processed into a representation called the "2 1/2 D Sketch", and 3) the 2 1/2 D Sketch is used to produce and recognize object-centered three-dimensional descriptions. These three stages correspond roughly to 1) both feature extraction and feature aggregation, 2) the production of a surface interpretation, and 3) an object identity and

shape understanding interpretation.

In the "primal sketch" intensity discontinuities are gathered together and significant lines, edges, and their spatial relations are represented. Most scene analysis systems refer to this type of representation as a segmentation.

The "2 1/2 D Sketch" is free of identities of objects and therefore does not require specialized knowledge about particular objects. The shape and position of the surfaces (depth and orientation) are made explicit at this intermediate level. Because surface orientation and depth can be used to describe arbitrary shapes, this is an ideal level of representation that lies between segmentation and object understanding.

The third stage of processing results in the interpretation of the scene as a composition of objects. Each object is described in terms of its shape and disposition with respect to the camera. This third stage of processing results in the identification of objects, and fits our description of the term "interpretation".

Didday and Arbib developed a visual system model that explains perceptual aspects of animal behavior (Didday 1973). In this system, a model of the current world situation (commonly referred to as "short term memory") is built from stimulus input and associations that can be drawn from stored experiences ("long term memory").

Didday and Arbib's work support the use of a dual visual system employing both peripheral and foveal components. The peripheral subsystem is responsible for discovering unexpected change. The foveal subsystem is steered to investigate areas of the visual field that demand attention by the resolution of competition between unresolved elements in the representation and the unexpected changes in the periphery. Through this construct, the goals of the perception facilitate perception of scene components required for survival by influencing the appropriate action-oriented elements of the internal representation.



II.3.2 General vs. specialized systems. As we make the transition from theoretical visual systems to implemented systems that embody limited amounts of generality, we should consider the trade-offs between general and specialized systems.

Many successes in scene analysis have been in very limited or specialized areas where ad hoc structures and processes have delivered good results. Such systems are noticeably rigid in their implementation, cannot be easily reformulated to act in other domains, and do not offer the research field much understanding of how to solve general vision problems. Bullock shows that the designs of general systems are sophisticated, and in order to accomplish a wide variety of tasks they are sub-optimal in solving any one task (Bullock 1978). He describes three types of general vision systems called matching, cueing, and interpretation. Basically, all three types derive an intermediate representation through feature analysis, and then by matching the intermediate representation with stored object models, the name, location, and identities of objects are derived.

Rather than a goal of generality in vision systems, Bullock argues for flexible configurations of specialized subsystems that closely match their domains and achieve optimal solutions. Hopefully, such an approach will result in systems that have more speed and accuracy, and are more practical to construct than general systems.

We have limited this review to some of the literature that is aimed at interpreting images of natural scenes in terms of the objects that appear. Most systems use some form of image segmentation as an intermediate structure. The identity of objects is either derived from the segmentation, the segmentation is derived from the object models, or the segmentation directs the application of verification programs that confirm identities from the image data directly. System design differences should then be viewed as differences in the methods that bring together semantics (object models) and syntax (image and feature values) image interpretation.

II.3.3 Blocks world scenes. Roberts designed and implemented a system which attempted to automatically locate and identify objects from image information (Roberts 1965). His work pioneered the field of computer

image understanding. The task domain was indoor scenes of blocks (a small set of polyhedra), and the goal was to locate and identify the block types.

In the first of a two stage process (see figure 8), spatial discontinuities of intensities were located in the image. These discontinuities in intensity were assumed to be caused by the boundaries of surfaces in the scene. Thus, the lines formed by simple inter-pixel intensity differencing were likened to a line drawing of the object.

Then, a set of three-dimensional atomic object models were compared with the lines extracted from the image. A suitable match was sufficient for the recognition of the object. Although the system correctly identified objects in many cases, it performed poorly in others. Problems occurred where shadows, missing lines, and strong intensity gradients were present.

Eventually this work was extended in two important ways by others (Shirai 1973, Waltz 1972). By including shadows as part of the model of the objects, and also by improved line extraction, these latter systems became very good at recognizing polyhedral scenes. Although it is not clear that this approach can be extended into the

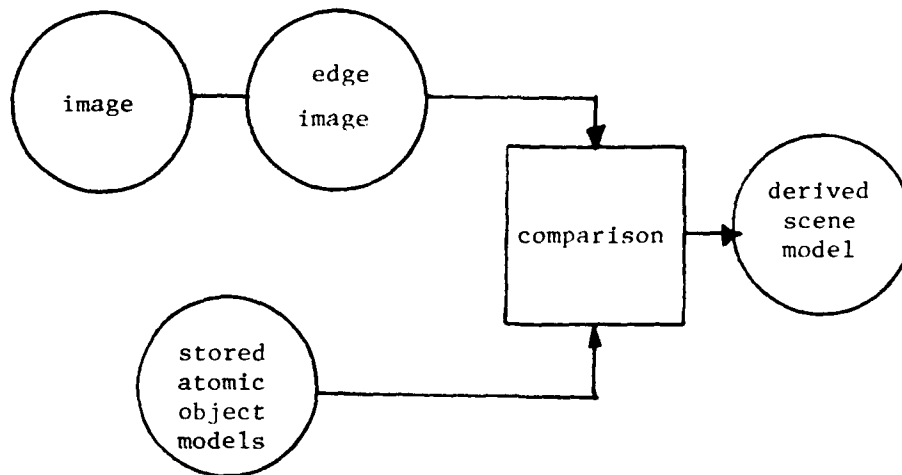


Figure 8. Roberts' system for blocks-world analysis

non-polyhedral world, early research systems did demonstrate the application of object model information to the image understanding problem.

II.3.4 Real world scenes. Yakimovsky and Feldman (Yakimovsky 1973), and Tenenbaum and Barrow (Tenenbaum 1973 and 1976), have integrated the interpretation and the segmentation processes (see figure 9). These scene analysis systems contain information in the world model that is used to permit or block the joining of pixels into regions. Through a succession of pixel joinings, the image is segmented into objects, where object identification is based on color and image position. The world model is in the form of likelihoods of adjacency between all pairs of objects.

Yakimovsky's segmentation proceeds by joining pixels and regions through a decision tree analysis, while Tenenbaum's process uses an iterative technique that makes a partial interpretation and suggests joins (or blocks the joining process) between pairs of regions. Through the use of heuristics, both systems attempt to derive a segmentation that reflects the maximum likelihood decisions for the image, based on properties of objects and a priori probabilities for the set of

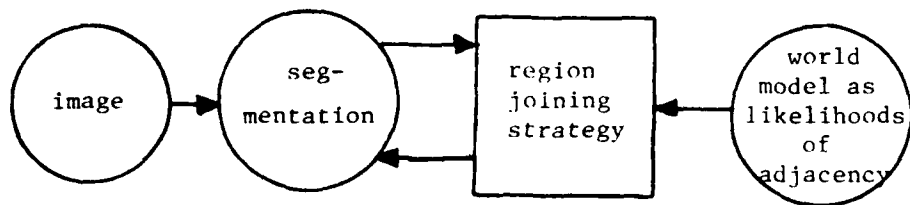


Figure 9. Interpretation guided segmentation systems

objects in the domain. Unfortunately, it is not always possible to obtain adjacency probabilities for the set of all objects in the domain. Also, it is not clear that adjacency in the image plane is a good measure for scene domains in which objects are likely to appear anywhere.

These systems show that top-down analysis is possible, where a model is used to "find" the best segmentation of the scene (directly in terms of objects). As Marr points out, the segmentation of the image directly into objects is only possible when specialized knowledge can be brought to bear (Marr 1977).

Tenenbaum and Barrow found it inappropriate to consider segmentation and interpretation as distinct processes (Tenenbaum 1975). They supported this with the observation that data directed (bottom-up) processing maintains analyses that depict the actual scene, but run into a bottle-neck of having too many possibilities to consider for interpretation (Barrow 1975). Processing which is entirely goal-directed (top-down) considers only the relevant possibilities, ignoring the unexpected, and produces results that do not depict the scene faithfully. They conclude that a combination of top-down and bottom-up processing is a good solution, but at that time

very little experimentation with such a paradigm had been done. In more recent work they discuss a framework for first generating a surface segmentation directly from image information before proceeding to interpret or name objects (Barrow 1978).

II.3.5 Query-directed systems. Several scene analysis systems that rely heavily on goal-directed analysis, or top-down processing, have been proposed and successfully implemented. Object or scene models are used to discover instances of objects, either directly in the image, or through an intermediate symbolic structure.

Garvey designed a system that finds particular objects in images when asked to do so (Garvey 1976) (see figure 10). When queried about an object, such as a chair in an office scene, the system produces a cost effective sequence of tests called a "strategy". The tests determine the existence and position of the object in the image. Advantages of this system are its ability to formulate strategies according to the goals of the user, and the ease with which new objects and object characteristics could be added. This system demonstrates the "test" step of a hypothesize-test paradigm. Once the human hypothesizes an object, Garvey shows that there is



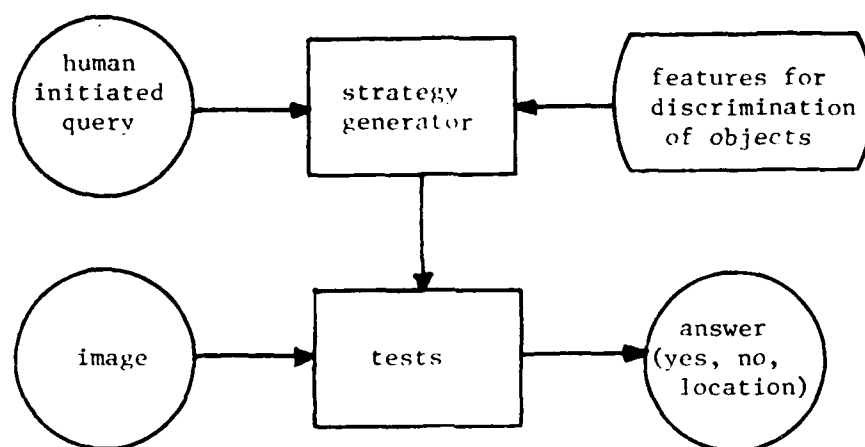


Figure 10. Garvey's query directed analysis

a method for preparing an optimal strategy for locating an instance of it.

Bajcsy and Lieberman use a model of objects and expected context to direct the analysis of natural outdoor scenes (Bajcsy 1974). First, the scene is segmented into regions based on inter-pixel similarity of color. Then, a strategy that uses a semantic network as a model of outdoor scenes locates the objects. The order of tests that the strategy applies is fixed.

This work is perhaps the first report of success at using object model information stored in a semantic network for top-down outdoor scene analysis. A more recent paper by Rosenthal and Bajcsy (Rosenthal 1978) proposes a structure for linking abstraction levels of a semantic network to resolution levels of an image so that queries about satellite images can be answered. They posit that, regardless of a choice of control structure, a hierarchy of visual knowledge is necessary to recognize objects in a context. In this system a query generates a sequence of objects to be identified before the queried object is located. Thus, a context of objects is generated. The identification sequence is derived from a hierarchy of objects that relate object size to

resolution of the image. Starting at the largest object first (coarsest resolution), the queried object is quickly located.

Although this technique of relating object size to image resolution is effective in satellite images, it is of doubtful utility in terrestrial images. Perhaps the sky and ground could be extracted at a coarse level, and all other objects at a finer level. We find that the use of partitioned semantic networks, where levels of abstraction form a hierarchy of visual knowledge is appealing as a structure for storing a priori information (see chapter V). Partitioning by abstraction level rather than resolution level condenses information that has similar use into identifiable areas, and allows a clear description of the relationships between elements of different abstraction levels (Williams 1977). However, we do believe that there is a need to understand the appearance of objects at different distances, and this can relate to image resolution.

Ballard, Brown and Feldman outline a method for developing a goal-directed vision system, where prior knowledge is used to guide the extraction of image descriptions (Ballard 1978). They use an intermediate

level called a "sketchmap" where a symbolic structure is built during analysis. This structure associates image elements with model elements, thus producing an interpretation. The model is a graph in which model nodes represent objects, and arcs between the nodes represent conditions where a relation holds. Also, procedures are described that instantiate model nodes into the sketchmap.

This system has been shown useful for answering queries in diversified areas of scene analysis, such as finding docked ships in aerial photos, and ribs in very noisy x-ray images. This intermediate structure lacks the surface level of abstraction that is necessary for general visual problems in outdoor scene analysis. The sketchmap is an instantiation of model information that can answer questions about an image. To make such a system general-purpose we would suggest that the first question should be "Describe all the surfaces in the scene.", and should be followed by, "Where are the objects?", and "What are their identities?".

II.3.6 General-purpose systems. Bullock describes general-purpose computer vision systems as they are applied to outdoor images (Bullock 1976a and 1976b). One

of the goals is to derive useful information from scenes, such as the identity, position, and description of objects. To produce results, a simplified implementation of a general design was used. In it a perfect model of each object is used, and only one object is found at a time. The system consisted of a matcher that compares models extracted from a sensor image with models extracted from a goal image. Thus, two intermediate structures are compared (see figure 11).

Considerable effort was expended in the selection of appropriate features for the model. It was shown that simple features, calculated on image data points, such as intensity, were easy to compute, but highly variable when objects were moved or lighting conditions changed. Global features that depict connected boundaries are very difficult to compute but are highly invariant for a given object. Features were chosen that fall between the two extremes, and their geometric relationships were used as a representation for comparison.

This system did not abstract surface models from images before proceeding to match against stored object information. It relies on the pre-specification of medium complexity features (edges) for the objects of

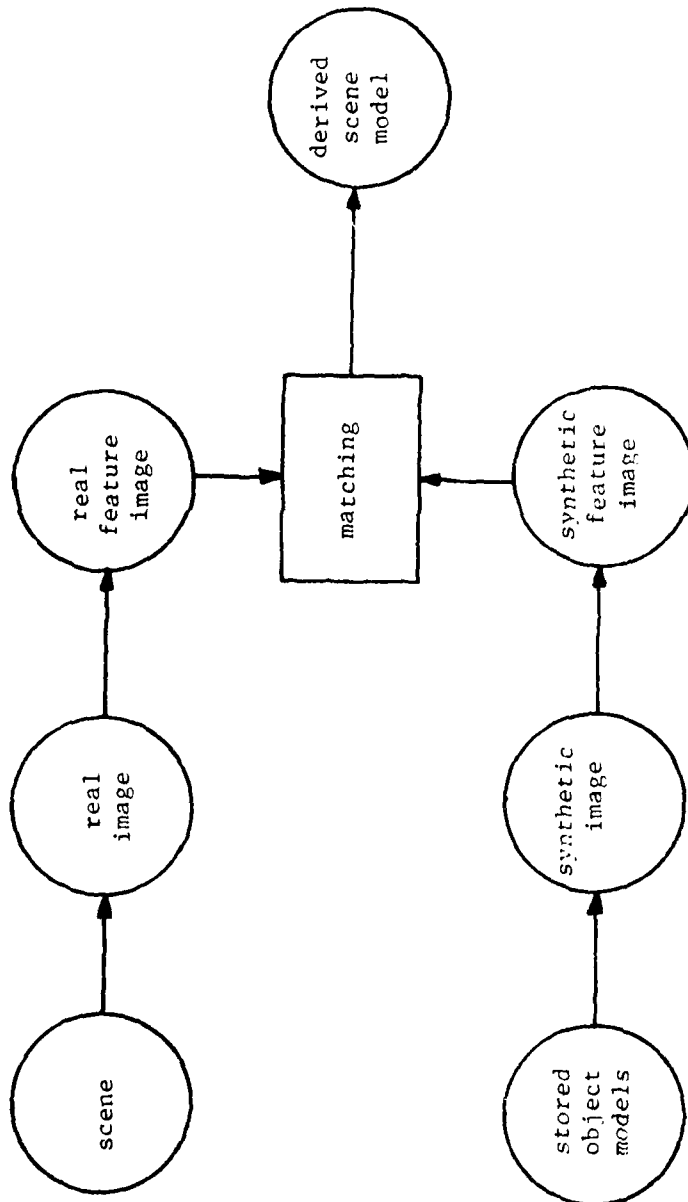


Figure 11. Bullock demonstrates a system that compares intermediate structures

interest. Although this approach works for certain objects and scenes, we feel that a matching process that relies on edge features can easily be overloaded by the myriad of edges and textures that are common to images of natural scenes.

In an evolving system called VISIONS, Hanson, Riseman and Williams (Hanson 1975, 1976, 1978a and 1978b, Riseman 1977, Williams 1977b) present a two stage approach to computer interpretation of images from outdoor scenes (see figure 12).

In the first stage, the image is segmented either by region or edge analysis. In region analysis adjacent pixels that have similar point features (Nagin 1979) are joined together. In contrast, edge analysis identifies dissimilarities in features of adjacent pixels, and collects them into region boundaries (Prager 1979, Hanson 1980).

The result of the first stage of processing is an image segmentation, coded into a graph structure. This graph is topologically similar to the regions, line segments, and vertices as they appear in the segmentation. Each region, segment, and vertex is represented by a node, and the nodes are connected by

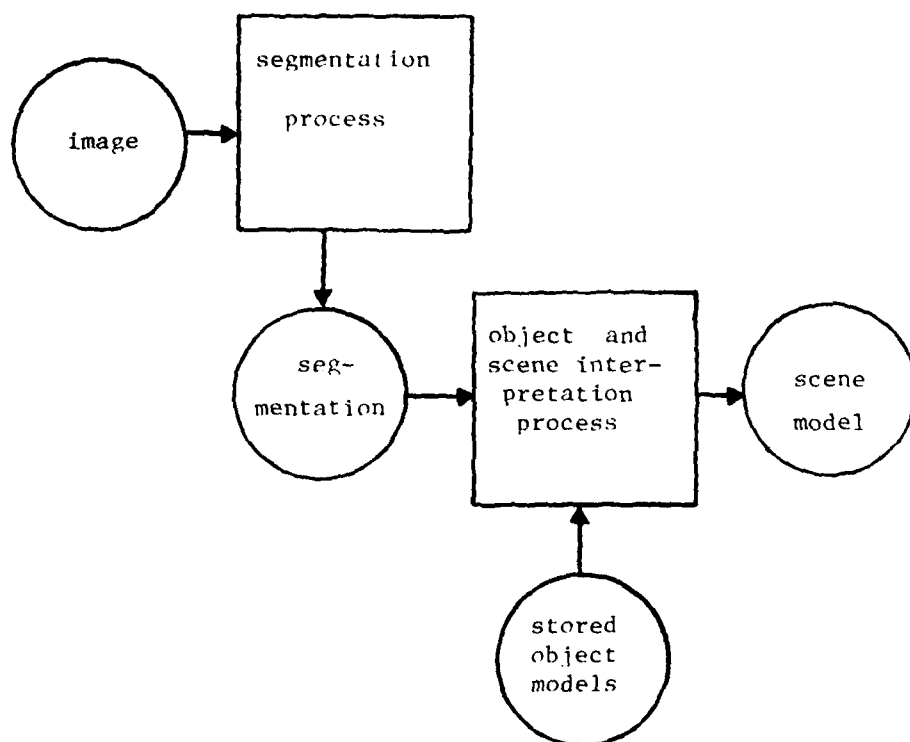


Figure 12. The VISIONS system



arcs that relate the regions to their bounding segments, and the segments to their common vertices. Values associated with the nodes in the graph describe attributes of the regions and line segments. Color, area, and centroid are examples of region attributes, and contrast and length are examples of line segment attributes.

The segmentation does not correspond to a surface interpretation. It serves as an intermediate structure in which there is a high degree of correspondence between some of the line segments and real surface boundaries, and therefore, also between regions and surfaces.

The second stage of the process is designed to interpret the segmentation in terms of surfaces, objects, and collections of objects. The problem of interpretation is divided into three components. They are the knowledge base, the processes that act on the knowledge base, and the strategy which applies the processes.

More recent activity in the VISIONS group (Riseman 1980) include the matching of three dimensional models to segmentations, and the inclusion of a feedback path, whereby partially instantiated interpretations can

influence the segmentation process.

The system presented in this thesis uses portions of the VISIONS system. The segmentation process is used to generate an initial model as described in chapter III. The object interpretation process uses the same structure as VISIONS and implements two processes in that structure. Chapter V details the object interpretation subsystem, examining both the data structure and the processes.

II.3.7 Static scene analysis summary. Static scene analysis systems that are intended to function on images of real outdoor scenes have at least one major problem to solve. That problem is the generation of a surface interpretation while making as few a priori assumptions about the scene as possible. On the path toward this goal various systems have been developed. They have been used to explore the application of pre-specified knowledge about the expected scenes. Few have attempted to build general purpose systems, but those who have show a multi-stage system that makes use of information both bottom-up (from the image to descriptions) and top-down (from the descriptions to the image).

#### II.4 Moving Image Analysis Systems

Many moving image analyses have been aimed at solving the correspondence problem, i.e., to determine the displacement of imaged scene points between two views. These analyses are usually applied in domains where all image points from any single object show motion components in a plane parallel to the image, such as cloud movements as viewed from a weather satellite. In other research, three-dimensional motion is analysed, but in very few scenarios is the camera in motion.

One important aspect of camera motion is occlusion. An observer, viewing a movie of our scene, can identify places in the scene where objects occlude one another. Occlusion is caused by the presence of a nearer object between the viewer and a more distant object.

The effect of occlusion in moving images has received attention in image analysis because many motion understanding systems employ correspondence techniques. The problem is that inter-image differences due to motion (displacement of an image component) must be distinguished from inter-image differences that are caused by occlusion and "disocclusion" of scene surfaces (see figure 13). In this figure the lack of correct

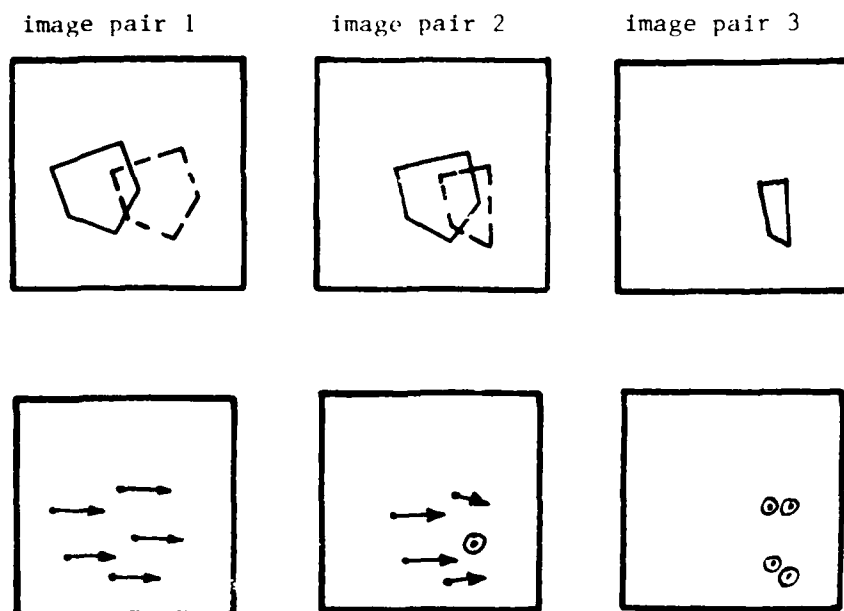


Figure 13. This figure represents the failure of a correspondence system during occlusion. The image pairs indicate solid outlines for the  $t_0$  image and dashed outlines for the  $t_1$  image. The object is being occluded as it passes behind a vertically oriented (invisible) object. In the correspondences, arrows signify corresponding points that would be discovered if vertices were being matched. Circled dots represent points visible in only one image for which only incorrect matches can be found.

inter-image matches is created by the effect of occlusion. As we discuss the motion analysis techniques, the effect of occlusion in each case will become apparent.

We will briefly examine moving image analysis as a research area divided into four sub-areas, according to the types of analysis performed. The sub-areas chosen are vector field, tracking, predictive modeling, and relaxation analyses. This examination is followed by a summary of the salient attributes of these systems as they apply to the determination of depth in the complex domain of moving images from real world scenes.

II.4.1 Vector field techniques. In vector field analysis, the origins of vectors are fixed (one each) to a number of points in the first image (see figure 14). The goal is to discover the end points so that each vector represents the spatial displacement of a local image feature between the first and second image (Ullman 1978). This discovery process is usually automated by a search that relies on a similarity measure (also called a "matching function" (Burt 1976)).

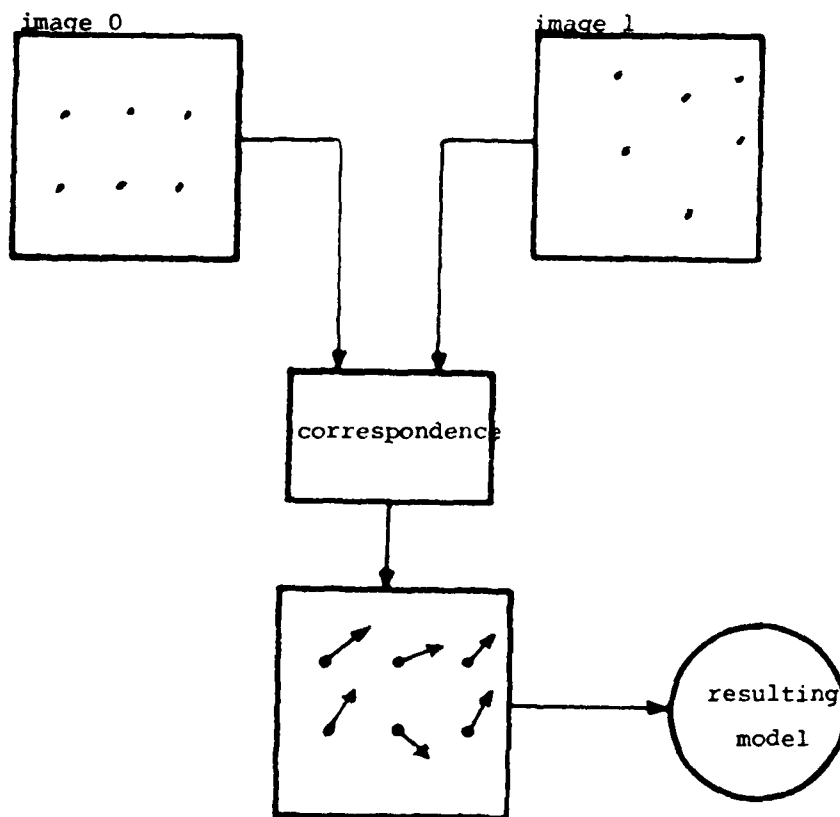


Figure 14. Vector field analyses solve the correspondence problem by treating the displacements as a field of vectors.

The resulting field of displacements can be analysed (with respect to a camera model) to infer the existence of rigid bodies, and in some cases a three-dimensional representation is derived. Gibson (Gibson 1966) calls the fields "optic flow", and Lee (Lee 1974), Cloksin (Cloksin 1978) and Prager (Prager 1979) demonstrate some characteristics of the field that determine depth.

In strictly bottom-up applications, the vector which describes the displacement of a point can have any direction and amplitude. In some applications, constraints on the motion of scene points is available from knowledge of either the approximate disposition of scene surfaces, or the camera displacement (direction and magnitude of camera motion) between frames. Constraints on point displacements result in a restricted area of the image over which the search for a match needs to be conducted.

Quam pursued a change detection technique using correlation to achieve registration of a pair of images as taken from a satellite (Quam 1971). The registration was modeled as a set of polynomial functions. These functions were modified so that they would match the field of vectors obtained from a set of cross-correlation

measures. The original specification of the functions requires knowledge of the camera positions and the curvature of the object (the planet being photographed). Although this system was not intended to model motion, it does relate a three-dimensional model to points in an image pair.

Quam and Hannah demonstrate a system that automatically determines depth from pairs of satellite photographs of Mars (Quam 1974). With assumptions of little change in range or sun angle between the pair of photographs, the images of corresponding scene points are compared by using a cross-correlation measure. A model of the depth is computed from the resulting displacements. The model is displayed as a contour map of the planet's surface.

C. Thompson (Thompson 1975) improves the correlation technique of Hannah (Hannah 1974) by using some criteria for acceptance of a correlation match. Two of these criteria are tests which can reject many false matches. They are based both on similarity of variance between the areas surrounding the matched points in each image, and on the similarity of the correlation peak (between the two images) and the autocorrelation peak in the first



image. An autocorrelation array of values is obtained by cross-correlating the first image area with itself. The second test is equivalent to the question: is the match value obtained equal to the one obtained when the area is matched with itself? Furthermore, Thompson suggests a local search around the correlation peak to improve resolution of displacement.

Thompson then deals with objects that change in size or shape between the images, a phenomenon he terms "perspective distortion", i.e., the effect of three-dimensional translation of the object. If the angle of a surface (relative to the camera position) is known, then the search for a match between image points that lie on the surface can be directed according to an expected displacement. Thus, a reduction in search effort is possible if the surface orientation is known beforehand.

Nevatia shows that correlation can be used in a succession of movie frames (Nevatia 1976) to determine depth. The scene contains one object - a cup that is covered with dark wrinkled paper. An "interest" predicate chooses windows that are good candidates for correlation matching. Rather than computing the

correlation coefficient, he uses the mean square difference which was computationally more efficient, and produced adequate results.

The search for a match can be reduced by having a model of the camera-object motion between images. Because Nevatia used a single rotating object, all points move along curves in the images (see figure 15). By searching in the vicinity of these curves, the correspondence is quickly found. Integer (pixel) displacements are found between successive views, and the path of the point is interpolated across the sequence by fitting a hyperbolic arc. A three-dimensional model is then inferred for the points on the object that the interest predicate selected.

W. Thompson has recently demonstrated a system for segmenting an image pair based on contrast and motion (Thompson 1979). By using a technique developed by Limb and Murphy (Limb 1975) which was designed to reduce television data bandwidth, Thompson derives motion estimates from intensity gradient information. The spatial slope of intensity is measured around a point in one image, and the value of intensity at the same point in the second image is recorded (see figure 16). The

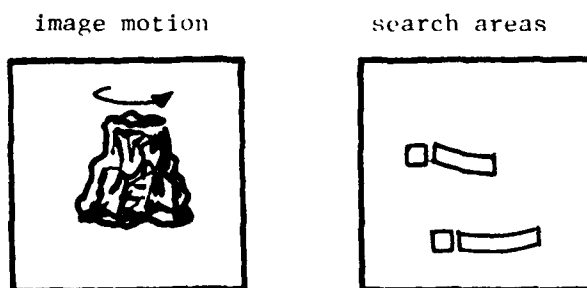


Figure 15. Because Nevatia had a model of the motion of the object, he could predict the approximate displacement area for matching. The square areas were searched for in the adjacent curved areas in the succeeding frame. The object was an inverted cup with wrinkled paper covering its surface.

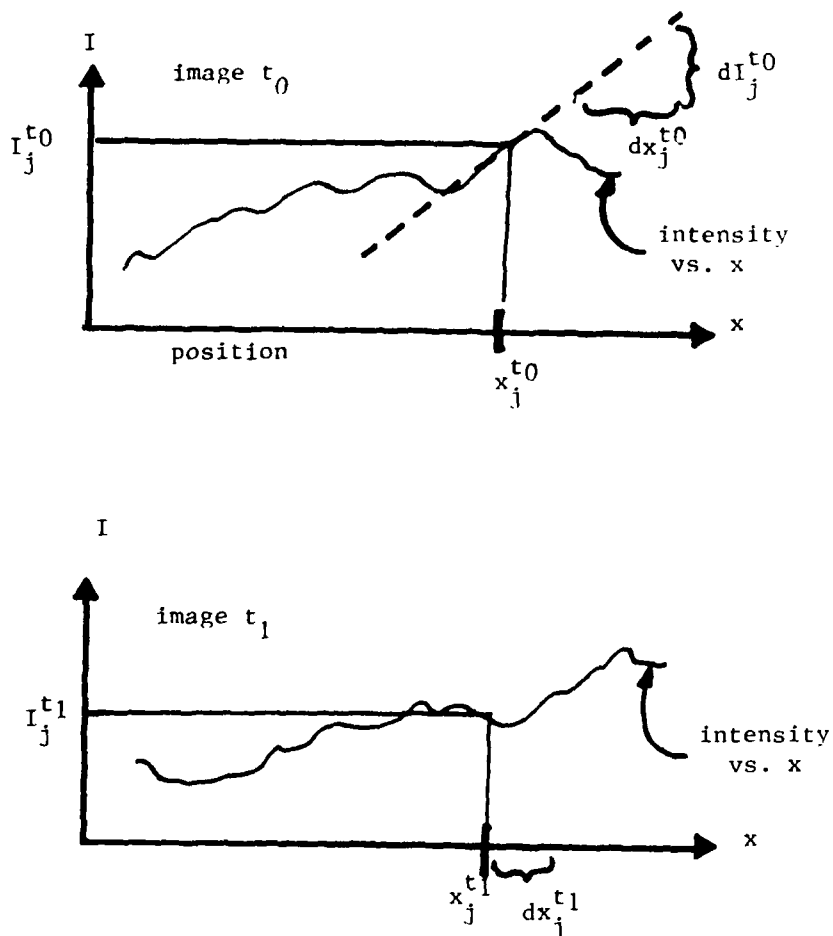


Figure 16. W. Thompson derived estimates for the motion of each point. First, the slope of intensity in the first image was measured around the point and the intensity value is recorded. Then the expected displacement is computed:  $dx_j^{t_1} = \frac{(I_j^{t_0} - I_j^{t_1}) dx_j^{t_0}}{dI_j^{t_0}}$ .

spatial displacement that places the second intensity value on the intensity slope of the first image is used as a local displacement estimate.

A set of global displacement estimates is then formed by collecting the local estimates in a Hough transform. Peaks in the transform space correspond to frequently occurring motion vectors, i.e., image areas that move together. Each local estimate is then replaced by the global estimate which it most closely matches. The resulting vector field is then used in the segmentation process.

The segmentation process first considers areas of strong gradient to form regions, under the assumption the image edges often correspond to surface edges. Then adjacent regions containing similar displacement vectors are merged, and any region covering different displacements can be split. After splitting and merging, a final segmentation is formed.

An advantage to this bottom-up system is that it does not perform a search, either locally or globally. Also, this system responds to both static and dynamic pictorial cues, perhaps easing the incorporation of its techniques into existing static analysis systems.

Unfortunately, the process of identifying sets of points as separate objects, based on the similarity of their displacement vectors, rules out the possibility of arbitrary motion. With the camera in motion, displacements of image points from a single surface generally vary in both amplitude and direction. Additionally, with multiple objects there are likely to be many objects with similar, perhaps overlapping, distributions of displacement vectors.

II.4.2 Tracking techniques. Jain and Nagel demonstrate a system for extracting moving objects from television images when the camera is stationary (Jain 1978). The scene used is a street corner as viewed from above, and the moving objects are automobiles and pedestrians. The system does not have any prior knowledge of the type of scene or any models of objects.

This system first measures statistics of the intensity values across a pair of images. Then, inter-image differences of these statistics show areas of occlusion and disocclusion, and hence, the leading and trailing edges of all moving objects (see figure 17). These moving edges are used to compute the velocity and size of the moving objects. If the system is given

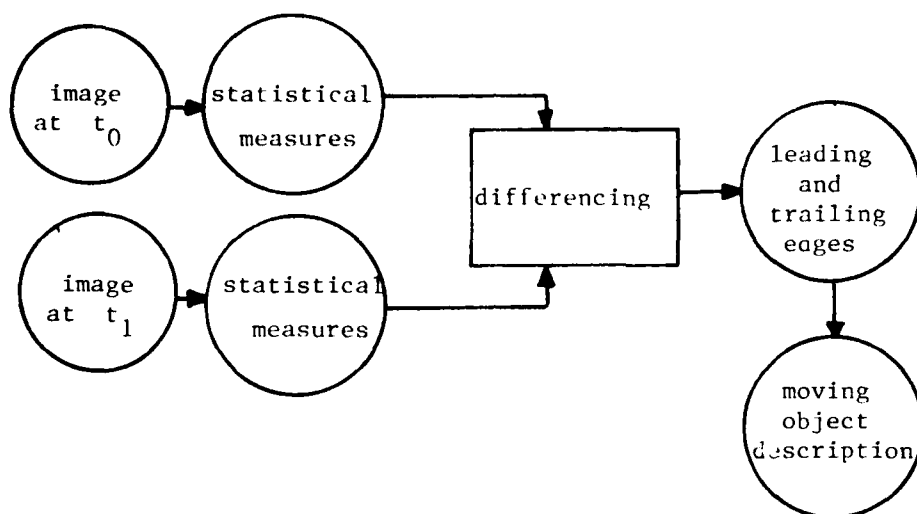


Figure 17. Differencing technique used by Nagel for moving object extraction

enough frames of the sequence (so that all moving objects travel at least their length) it can automatically remove the moving objects from one frame to produce a reference image of only the stationary scene components.

In another work extending this technique, Dreschler and Nagel extract the moving portion of television images where the camera is stationary (Dreschler 1978). Regions are produced by pointwise intensity differences between a pair of frames. These regions are then hypothesized as objects. A set of features is measured across each region to define a vector in multi-dimensional feature space. Because the features of an object change little between successive frames, a cluster of vectors is formed by one object from a sequence of images. These clusters are found through a minimal spanning tree search. Any particular object can then be found in each frame by mapping back from feature space to the images.

Radig demonstrates a region matching technique for tracking and describing moving objects in the same scenario (a stationary background) (Radig 1978). Features that represent the intensity value and the gradient of intensities are measured in each frame. These features are collected into regions of similarity



and compared with other regions in the preceeding and succeeding frames. Comparisons of these features effectively match the regions' internal structure when linking together regions through time.

Milgram developes a technique that tracks the moving image of military targets as they appear in infra-red images (Milgram 1977). In this domain, a threshold of the image is sufficient to separate the target from the background. The selection of the threshold is automated by searching for that threshold in each image which produces the most consistent match of the target size, shape, and expected position. A dynamic programming approach is employed for the search phase.

Price solves change detection problems in image pairs from a variety of scenes (Price 1976). The technique first segments both images independently, and then matches regions based on the similarity of their features. Adjacency of other regions influences the match process (see figure 18). The system is demonstrated on images from aerial and terrestrial scenes, as well as radar images.

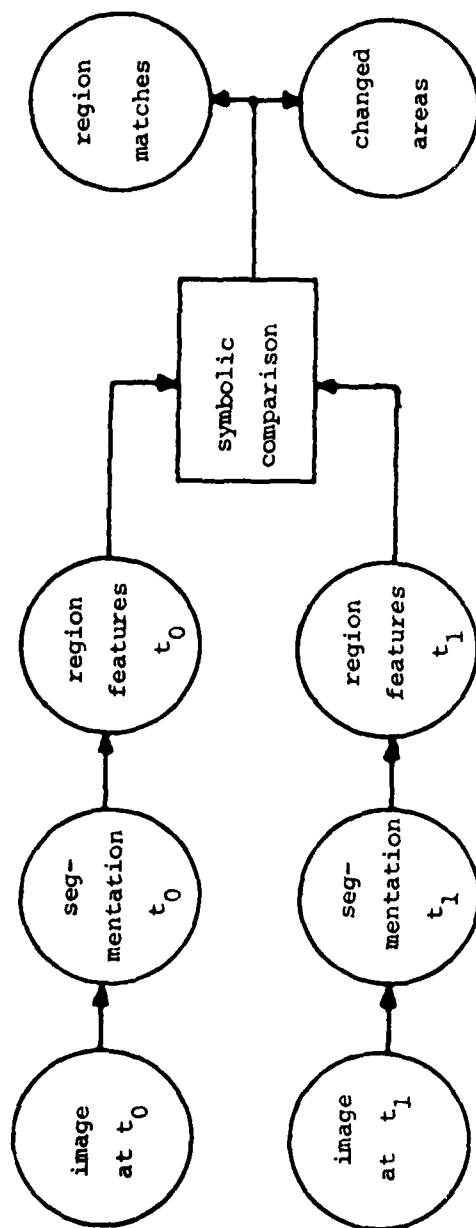


Figure 18. The change detection system developed by Price compares regions.

This symbolic matching technique is capable of finding correspondence of regions where there are many regions in each image. Changes in camera position, or in the positions of an object could be handled easily, and mis-matches of large regions were infrequent. The technique was shown to be faster than cross-correlation and better than image differencing techniques for the problem of change detection.

Let us consider the utility of several of the techniques in our problem of motion analysis. From the work on tracking we see that simple inter-image differencing techniques are adequate for detecting moving portions of images. When the camera is moving through the environment however, all portions of the image, except for very distant objects, are in motion.

Thresholding was discarded as a surface or object extraction method in this thesis because, in real world scenes, simple thresholding does not extract object boundaries. We do employ a version of tracking that corrects for errors in our system by forming regions from difference images. This is explained in chapter III, section III.2.5.

Comparison of two segmentations is probably not a suitable method for motion analysis. Small regions change very rapidly between images of real terrestrial scenes. They are quite often near the edges of surfaces where resolution of displacement is most critical. If small regions are not collected into the surfaces they are expected to represent, then measurement of region displacement is grossly inaccurate and, thus, inadequate for determination of depth.

II.4.3 Predictive modeling. Martin and Aggarwal examine several dynamic scene analysis systems, and demonstrate a system that extracts objects from images of scenes where the objects display motion in a plane parallel to the image (Martin 1977). This system attacks the problem of modeling objects where considerable occlusion is taking place.

To handle occlusion, a predictive model is generated. This model is formed from the boundaries in the binary images that are given to the system as input. Descriptions of the boundaries are matched between frames to produce a vector field, and those vectors that cluster together are assumed to belong to a single object. Once the model of objects is formed, the search for

correspondence of boundaries in successive frames is reduced through the model's prediction of displacements. Areas where occlusion and disocclusion are taking place are also predicted by the model (see figure 19).

This system is extended by Roach and Aggarwal (Roach 1979) to three-dimensional polyhedral bodies with arbitrary motion. In this work, internal edges of polyhedra are also tracked as part of an object.

These systems require "clean" edge features (the binary input images), which are not directly available from real outdoor images. The possibility of applying the predictive modeling approach to real image analysis should not be discounted however. In a system that deals with real images, the primary obstacle to using this system is in its reliance on edge features for description and matching. The use of a model that predicts image dynamics is essential to the system presented in this thesis (see chapter III).

II.4.4 Relaxation techniques. Rosenfeld outlined a relaxation technique that would form a cubic lattice of pixels, where  $x$ ,  $y$ , and  $t$  are the orthogonal dimensions (Rosenfeld 1979). An update rule would then collect volumes (rather than regions) of similarity,

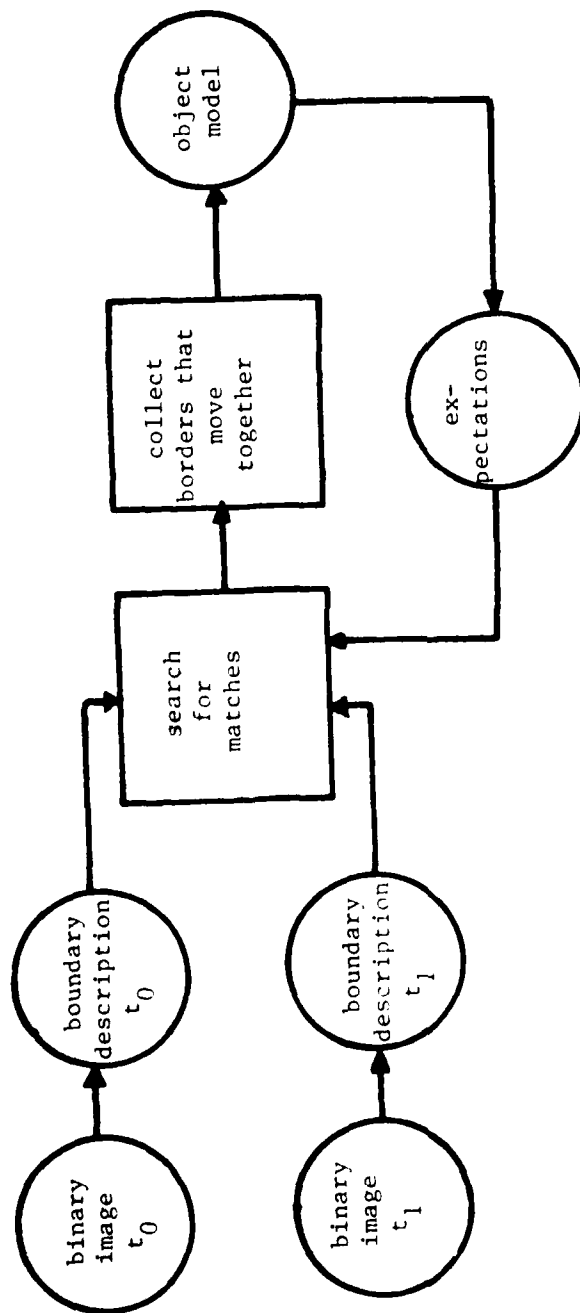


Figure 19. System by Martin and Aggarwal uses the derived model to predict occlusion and reduce the search for matches

simultaneously segmenting the dynamic image both spatially and temporally. This approach would still involve search for the appropriate correspondence between temporally disparate pixels.

Prager effectively solves the correspondence problem for synthetic images where arbitrary motion of a solid object and the observer occur (Prager 1979). The process first produces edges using inter-pixel differencing, and codes these edges according to the direction of the local intensity gradient.

Displacement vectors are associated with each point in the first image, and a relaxation process is defined where each vector is iteratively updated. The update is based on the match between the feature at the base of the vector in the first image and the features near the tip of the vector in the second image. Also, the similarity of vectors in the neighborhood of each vector influences the update. The process is iteratively applied until all vectors have suitable matches between their base and tip. The resulting vector field can then be segmented to determine the rigid bodies of the scene.

Among the advantages of this approach are a high degree of local parallelism, and the ability to deal with motions in the third dimension. Also, the system could handle multiple objects. Although this system was tested on a real image sequence (the same sequence on which the system presented in this thesis is tested), the results were not as convincing as those for synthetic data. Anomalies in the data, and the restriction of edge placement to integer pixel positions are likely to have caused most of the problems with the use of real data.

II.4.5 Moving image analysis summary. The utility of edge features in motion analysis should be examined. Indeed, the internal visual structure of an object should exhibit properties that permit computation of accurate and meaningful displacement vectors. However, this is not necessarily the case for edges that are found on the outer boundary of an object.

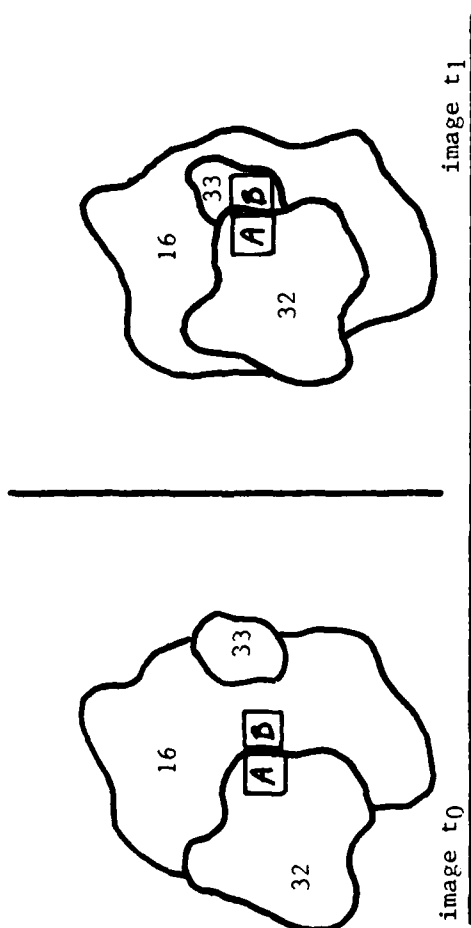
Edge and texture features are often employed in motion analysis systems because they supposedly indicate areas of the image that can be tracked with less ambiguity than point intensities. Edges are found between areas of different intensity, and these differences arise from a variety of phenomena. Because



different surfaces often have different image intensities, edges often will be found between the imaged points from occluding surfaces.

It should be obvious that the edge of an occluding region is an entity that belongs to and moves with the occluding region. Unfortunately, the response of an edge operator is dependent on the intensity difference across a boundary, between both the occluding and occluded regions. Thus, for a system to search for an edge in a new frame that corresponds to one in the present frame, it cannot rely on the discovery of an edge that has the same magnitude and form as the edge in the present frame (see figure 20). Rather, one must consider a scheme that would predict the response, i.e., it would have to either predict the intensity context around the position of the occluded edge in the new frame, or assume that the edge value would not change.

Prediction cannot be done unless the system has a model of either the surfaces, or at least the distances to points on either side of the edge feature. The same problem exists with the use of correlation techniques, but even more so because correlation measures are maximum when the pictorial structure is identical in both frames.



feature function :  $R = A - B$

$$R^{t0} = 16$$

$$R^{t1} = -1$$

Figure 20. One problem with matching interpixel features is that the response of the feature operator may drastically change. In this figure numbers on the regions indicate their average intensity.

Since these systems are aimed at detecting image dynamics, they do not have a model until the motion is detected. Therefore, edge features (or correlation) should only be used if there is some way of determining whether an edge indicates a discontinuity in intensity on a surface, or between surfaces, or else a way of ensuring that problems due to occluding surfaces has not interfered with the matching process.

Higher-order edge features, such as line segments and their points of intersections, called vertices, can also show variability during motion. This is due to the facts that such aggregates are collections of edge features with their own variabilities, and occlusion effects can cause motion of a vertex that is not related in a simple way to the motions of the regions that it results from.

To understand the motion of an image vertex consider the effect obtained by imaging a pair of nearly vertical objects by a camera moving in a straight line (see figure 21). The slight tilt of the edge of a tree, and its intersection with the edge of a telephone pole will result in the imaging of a vertex that moves vertically, while the motion of the two regions and edges relative to

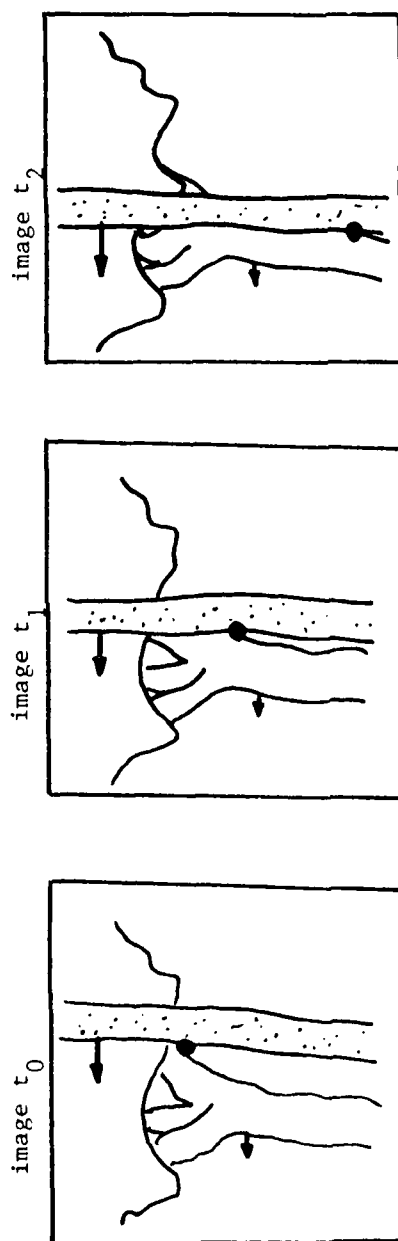


Figure 21. The two-dimensional motion of a vertex formed by a near telephone pole and a distant tree trunk is not related (in a simple fashion) to the motions of the participating surfaces or regions. In this figure the large dot indicates the motion of the vertex.

each other is purely horizontal.

Motion understanding systems that make use of vertices (Martin 1977b) must identify each vertex as belonging to either one surface or to the junction between two or more surfaces. We feel that this approach would be very difficult in our scene, since a large number of surfaces exhibit many occlusions.

Vector field analyses are an obvious mechanism for motion detection. They must be coupled with three-dimensional models, rather than simple clustering techniques, to obtain object descriptions where the camera is in motion.

Tracking techniques are useful where a stationary background is available, and simple image differencing is effective. We have chosen to adopt image-differencing as a method for resegmenting the image where an initial model is incorrect.

The use of a scene model is attractive for several reasons. Such a model could begin with little or no information, and through a refinement process, produce an improved model of the scene. Additionally, the use of a predictive model eliminates the problems of occlusion as

demonstrated by (Martin 1977 ). We classify such a refinement technique as an hypothesize-test system. The system presented in this thesis comprises image differencing, predictive modeling, and vector fields to produce a surface interpretation.

### CHAPTER III

#### THE SURFACE INTERPRETATION

If a person is shown a movie taken from the passenger seat of a moving automobile, he can report a perception of depth. The observer can predict when he will pass by objects, and can easily judge which is the closer of two objects. He can also judge the approximate orientation of a large planar object that exhibits a large distance gradient across its surface, such as the road. It is this perception of depth - distance to surface points - that we call the surface interpretation of the moving image.

An observer can bring several sources of knowledge to bear upon the sensory data in determining depth. Because the moving image is monocular, the viewer cannot make use of stereoscopic depth information. However, experience with the size of familiar objects can help with distance judgements. Additionally, cues available in static images can be used, such as texture gradients, shadow, and occlusion. The system presented in this thesis uses dynamic aspects of the image to judge depth, an analysis based entirely on input data.

The first section of this chapter introduces the coordinate system and camera model. Then, real world motion and depth are related to changes that occur over a succession of images taken from a moving camera. The representation used for the surface model is examined next. The first section ends with a discussion of the features selected, and the process for initialization of the surface model.

In the second section of this chapter a system is described which produces a surface interpretation. The surface interpretation process is a sequence of sub-processes that utilize a hypothesis-test strategy (see figure 22). The first sub-process is responsible for generating an initial model using static analysis. The second refines the model, obtaining distances to hypothesized surfaces. The final sub-process corrects the model by discovering portions of hypothesized surfaces that move in a manner suggesting that they are distinct (separate) surfaces.

The algorithms presented in this chapter are described pictorially as data-flow graphs. The convention adopted here uses arcs to represent data flow, circles to represent data, and boxes to represent



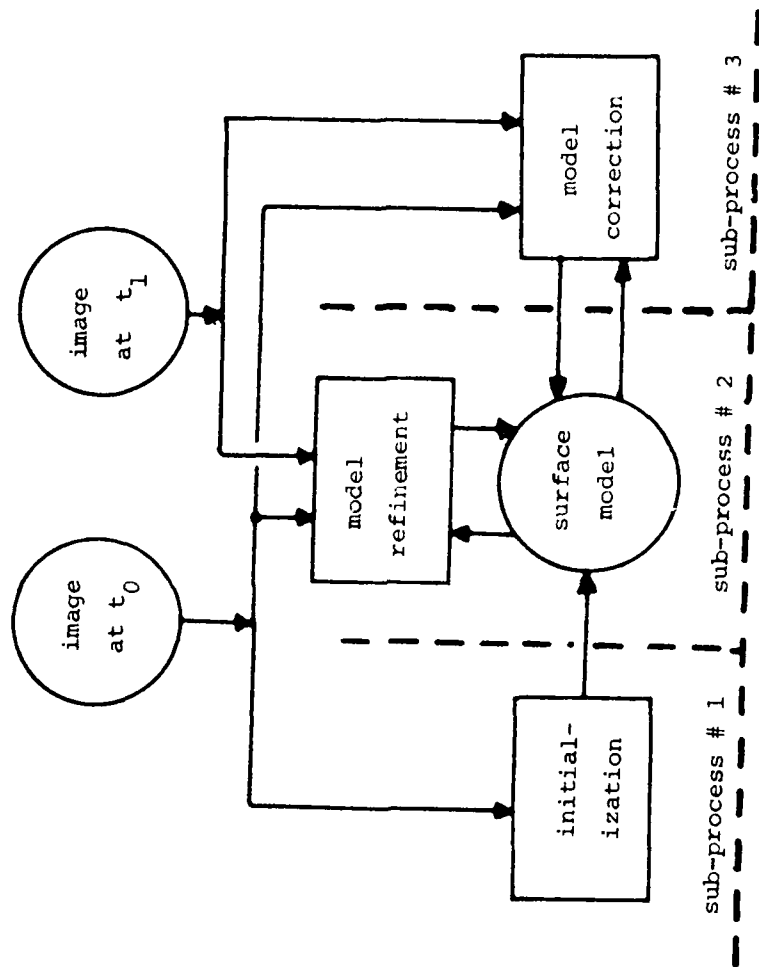


Figure 22. The three sub-processes for surface interpretation

AD-A107 565

MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INF--ETC F/G 9/2  
COMPUTER INTERPRETATION OF A DYNAMIC IMAGE FROM A MOVING VEHICL--ETC(U)  
MAY 81 T D WILLIAMS  
COINS-TR-81-22

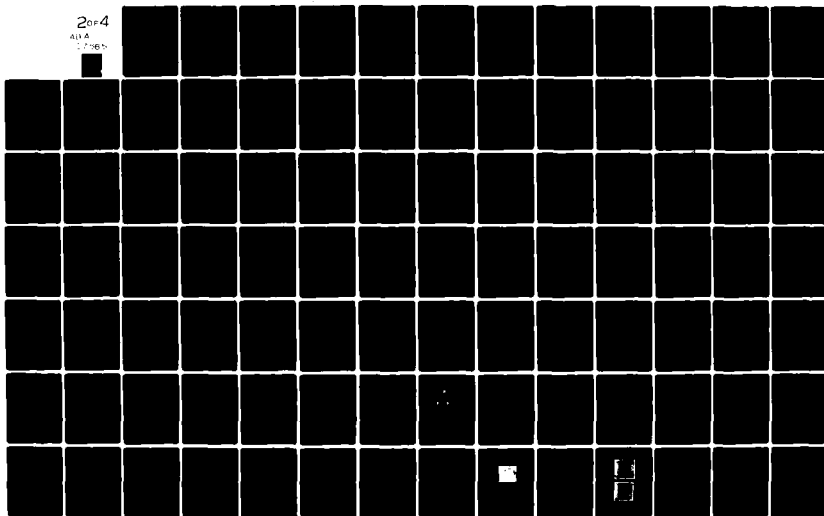
N00014-75-C-0459

NL

UNCLASSIFIED

204

ALL A  
17065



processes. From these descriptions it is easy to determine the amount of computation and memory that are required to realize the system.

### III.1 Representation and Issues

Before discussing the mechanisms for deriving a surface interpretation, we examine several issues of representation. We must specify the coordinate system and projection relations. From these we define the concept of "focus of expansion", a point which relates the direction of travel to the image coordinate system. Then, the representation of the surface model is described. As mentioned in chapter II, the selection of features for motion analysis is quite important. The features chosen are presented next, and finally, the technique for initialization of the surface model concludes this section.

#### III.1.1 The problems of "start-up" and "continuation."

The discussions that follow describe the problems and techniques for deriving a surface interpretation from a pair of images. The term "motion stereo" is commonly used to describe such analyses. We partition the set of successive movie frames into a sequence of image pairs (see figure 23). The earlier and later image of each

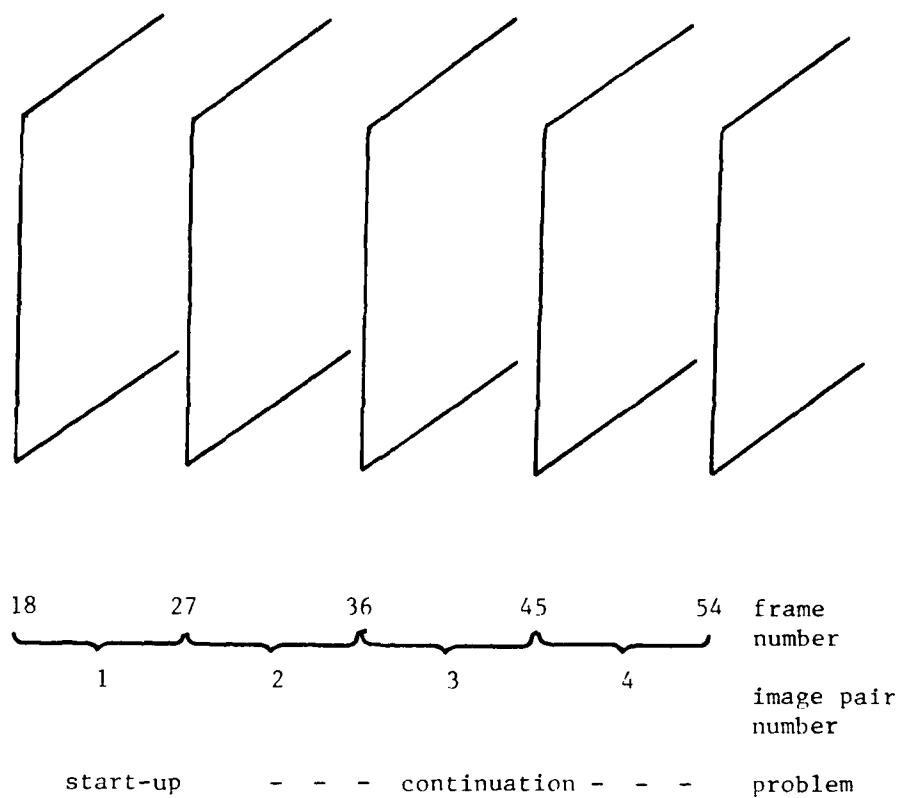


Figure 23. The sequence of selected frames is formed into a sequence of image pairs for analysis.

pair are referred to as the  $t_0$  image and the  $t_1$  image, respectively.

Derivation of the surface model from the first pair of the sequence is called the "start-up" problem, because the system has no internal representation of the scene when it begins analysis. After a model is derived (and refined) the next image pair can be used to (further) refine the existing model. Model refinement continues for all image pairs except for the first, and is called the "continuation" problem. This chapter is focused at solving the start-up problem because the solution of the continuation problem is believed to incorporate a subset of the analyses needed for start-up. This thesis does not provide a solution to the continuation problem. That is left for future work.

III.1.2 Moving image projection. In order for us to derive the positions of scene points from the positions and motions of the corresponding image points, we must understand the camera model and the process of dynamic projection. The three-dimensional coordinate system is fixed to the camera, with the origin at the focal point of the lens (see figure 24). Thus, the coordinate system is moving with respect to the scene, and is stationary

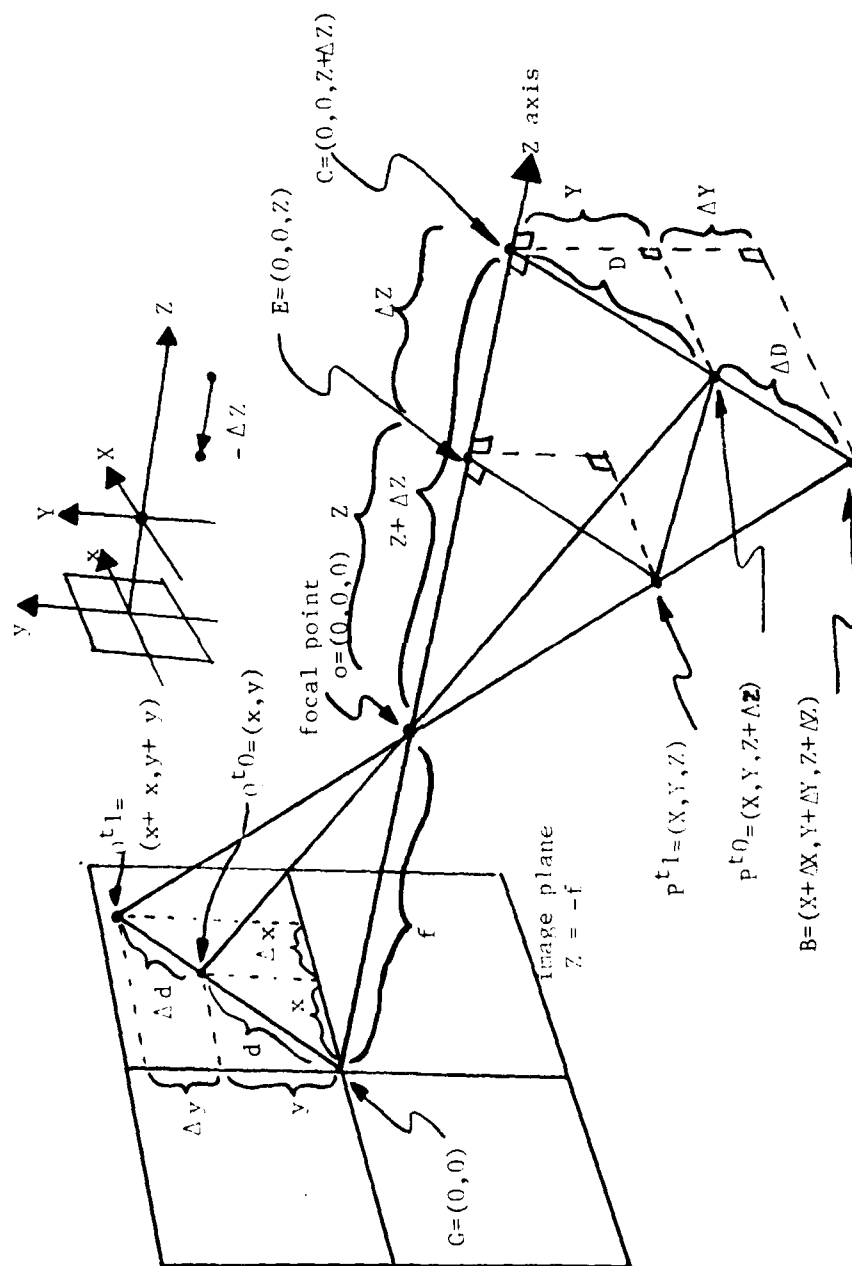


Figure 24. The coordinate system relating scene motion to image motion.

with respect to the camera. The X and Y axes of space are set parallel to the x and y axes of the film plane, and the Z axis points in the direction of travel. We use the capital letters X, Y, and Z to denote position in the environment, and small letters x and y for position in the image ( $Z = -f$ ) plane.

The X and Y axes represent perpendicular axes parallel to the x and y axes of the image plane. If the image is perpendicular to the direction of travel, all three axes are at right angles (as viewed from an ortho-normal three-space coordinate system). Otherwise, the X axis is rotated according to the pan angle, and the Y axis is rotated according to the tilt angle of the camera (see figure 25 and 26). For clarity, we will initially show an image plane perpendicular to the Z axis. Then, it will become obvious that the same relations that are derived for the case where the image plane is perpendicular to the Z axis will hold for the case where the X and Y axes are skewed relative to the Z axis.

The motion of the camera is defined as piecewise and rectilinear, i.e., between each frame the camera is assumed to be travelling in a straight line. Except for

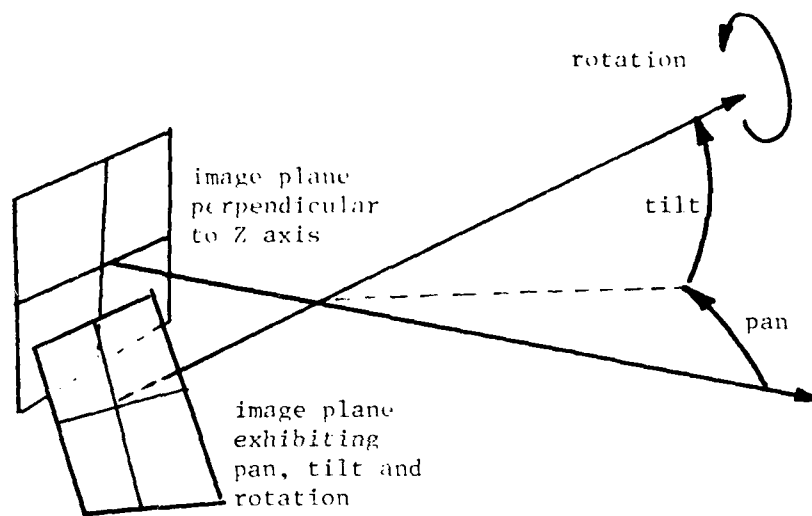


Figure 25. The three angles of the film plane



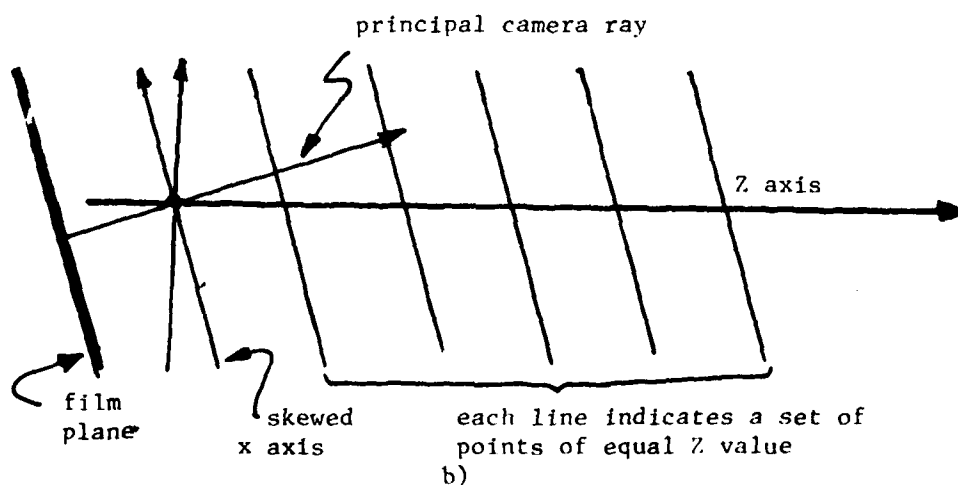
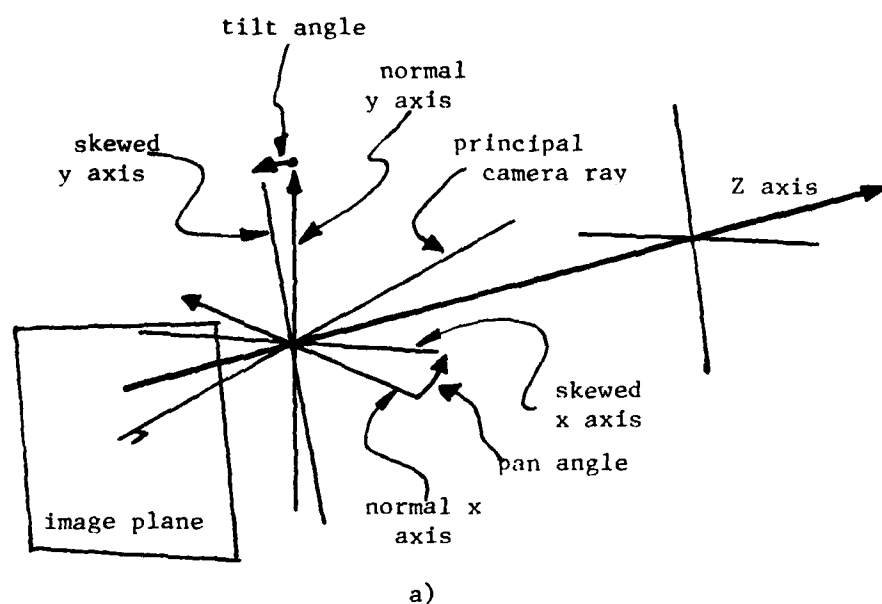


Figure 26. The pan and tilt angles define the skew of the x and y axes (a), and (b) a simplification by viewing along the y axis demonstrates the effect of a skewed x axis.

blur that occurs during exposure, there is no reason to assume that the motion is other than piecewise and rectilinear. The Z axis is defined by the line drawn between a pair of successive camera focal points. For clarity figure 24 is drawn with the coordinate system stationary, so all scene points are moving relative to the camera.

The only non-zero components of motion of scene points in the coordinate system are their Z components. We identify this motion component as  $\Delta Z$ . The term  $\Delta Z$  is obtained from measurements of the position of each movie frame exposure. The results of the entire surface interpretation process are scaled by this term, (as shown below) but the process itself is not affected by the accuracy of  $\Delta Z$ .

We use the letters P for scene points and Q for corresponding image points. P is a vector of dimension three (i.e., the coordinates in the scene are X, Y, and Z) while Q is a vector of dimension two (i.e., coordinates in the image are x and y) in the image plane ( $Z = -f$ ). Of course, it is also necessary to specify the time that scene and image points appear at particular locations.

Thus,  $P^{t_0}$  represents the position of scene point  $P$  at time  $t_0$ , while  $P^{t_1}$  represents the position of scene point  $P$  at  $t_1$ .

There are several other points that must be specified for the derivation of the relationship between image dynamics, positions of scene points, and camera motion.

In particular the position of the point  $B$ , the points  $C$  and  $E$  which are projections of  $P^{t_0}$  and  $P^{t_1}$  onto the  $Z$  axis, the origin,  $O = (0, 0, 0)$  at the camera focal point, and the image origin  $G = (0, 0)$  in the film plane are sufficient (see figure 24).

We are now ready to derive the new position  $Q^{t_1}$  of an image point given its old position  $Q^{t_0}$  and knowledge of the distance  $Z$  to the scene point.

By observation of similar triangles on the image plane we get:

$$\frac{\Delta d}{d} = \frac{\Delta x}{x} = \frac{\Delta y}{y} \quad (1)$$

Using similar triangles  $O E P^{t_1}$  and  $P^{t_1} P^{t_0} B$ , we

observe that:

$$\frac{\Delta D}{D} = \frac{\Delta Z}{Z} . \quad (2)$$

Since triangle  $O P^{t_0} B$  is similar to  $O Q^{t_0} Q^{t_1}$  and

triangle  $O C P^{t_0}$  is similar to  $O G Q^{t_0}$  we

observe that:

$$\frac{\Delta d}{d} = \frac{\Delta D}{D} . \quad (3)$$

Therefore by (1), (2), and (3),

$$\frac{\Delta x}{x} = \frac{\Delta y}{y} = \frac{\Delta d}{d} = \frac{\Delta D}{D} = \frac{\Delta Z}{Z} ,$$

$$\text{or} \quad \Delta x = \frac{\Delta Z x}{Z} , \quad \Delta y = \frac{\Delta Z y}{Z} , \quad (4)$$

and in vector notation the relationship can be expressed

$$\text{as} \quad Q^{t_1} = Q^{t_0} + \left( \frac{\Delta Z}{Z} Q^{t_0} \right) . \quad (5)$$

Thus, for the ideal camera model, the imaged displacement  $(\Delta x, \Delta y)$  for a point is easy to compute given some change in distance  $\Delta Z$  and a final distance  $Z$ .

Now, if we allow the camera to have a constant pan and tilt angle, and skew the coordinate system by these angles, the same relations hold. In figure 27 a simplified drawing shows the effect of one skewed axis. Similar triangles from figure 24 are still similar in figure 27. The major difference is that the Z axis intersects the image at a different point than the principal camera ray.

The use of a skewed coordinate system results in an interpretation that is derived under the assumption that surfaces of constant Z value are parallel to the image plane. In the case of the data employed in the experiments, the tilt and pan angles were very small and had very little effect on the system.

### III.1.3 Focus of expansion.

By observing any set of scene points over a time interval we will notice that all the image points  $Q_i^t$  that are projections of  $P_i^t$  move in straight lines outward from the image point O.

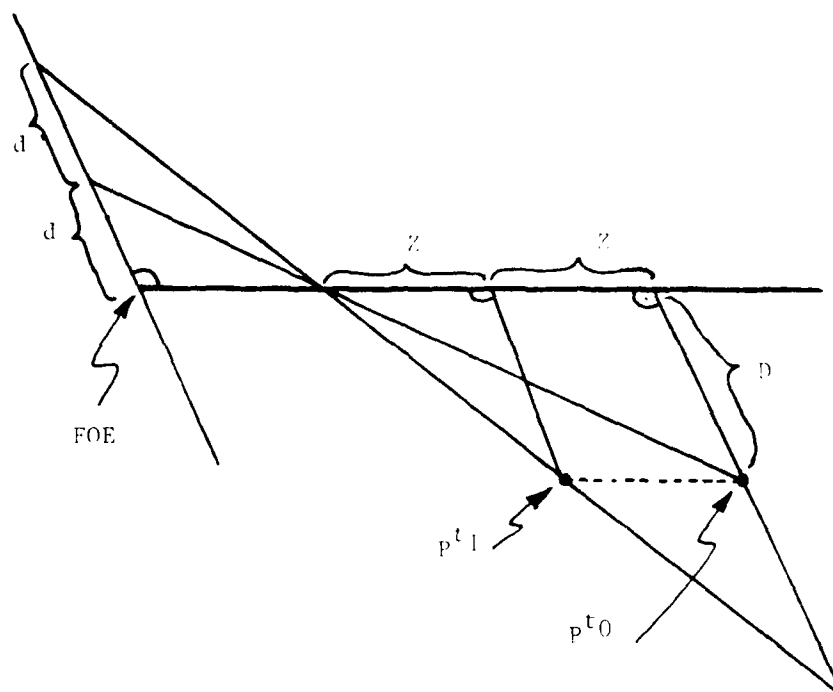


Figure 27. In the skewed coordinate system the same relations hold as those derived for the perpendicular system.

We define the central image point  $O$  as follows:

Definition: The Focus of Expansion is the point on the image plane intersected by the  $Z$  axis. The Focus of Expansion is abbreviated "FOE".

The FOE, defined by the scene's motion relative to the image plane, is the point at which the axis of travel intersects the image. This allows the direction of travel to be determined from the image dynamics. The values for  $x$  and  $y$  (the position of a point on the image) are measured with respect to the FOE. Since the entire surface interpretation process is based on relation (5), the accuracy of the FOE is quite important.

In images from a real camera the FOE may not lie at the center of the image or the principle camera ray projection point. Small changes in the image orientation can cause large changes in the placement of the FOE with respect to the image center. For these reasons, the FOE is not assumed to be at the center of the image. Rather, a process that is described in section two of this chapter searches for the FOE.

III.1.4 Assumption of non-rotating camera. One central assumption made for the camera model is that the film plane, and therefore the coordinate system, moves only in the direction of the Z axis. Put another way, the camera moves through the scene with a fixed orientation. It is nearly impossible to maintain the assumption in practice. Therefore, the images were registered to account for the major portion of the effect of small interframe changes in the camera orientation. Although this registration was done interactively, it could be accomplished automatically if the inertial frame of the camera were recorded along with the images.

First we will examine the effect produced by change in camera orientation, and then show the method of registration which accounts for most of the effect. Finally, we compute the resulting residual error.

The real camera has three orientation angles and a translational component that will fully describe its position and orientation. The three orientation angles are called "tilt", for the rotation about the X axis, "pan", for rotation about the Y axis, and "rotation", for rotation about the Z axis. The translational component is the spatial displacement of the focal point. The



translation of the focal point determines the Z axis, i.e. the axis of travel. Therefore, the ideal (model) and real translations are identical. The pan, tilt and rotation angles, however, are not accounted for in the model as described so far.

As shown in figure 28 the position of an image point on an actual (real) image can be related to the position that the point would have if the image plane were at some other (ideal) orientation. In order to compensate for the placement error that results from inter-image camera rotation we will formulate the difference between the actual and ideal positions to which a scene point would project. To simplify, we select the orientation of the ideal image plane to be perpendicular to the Z axis. Additionally, we simplify by showing a projection onto the xz plane, reducing the image plane to a line.

One needs only to compute the new intersection of the camera ray with the ideal image plane to determine the location that the scene point would project to on the ideal image. This can be done with the following information regarding the real image point: 1) its position on the real image, 2) the center of the real image, and 3) the position of the FOE on the real image.

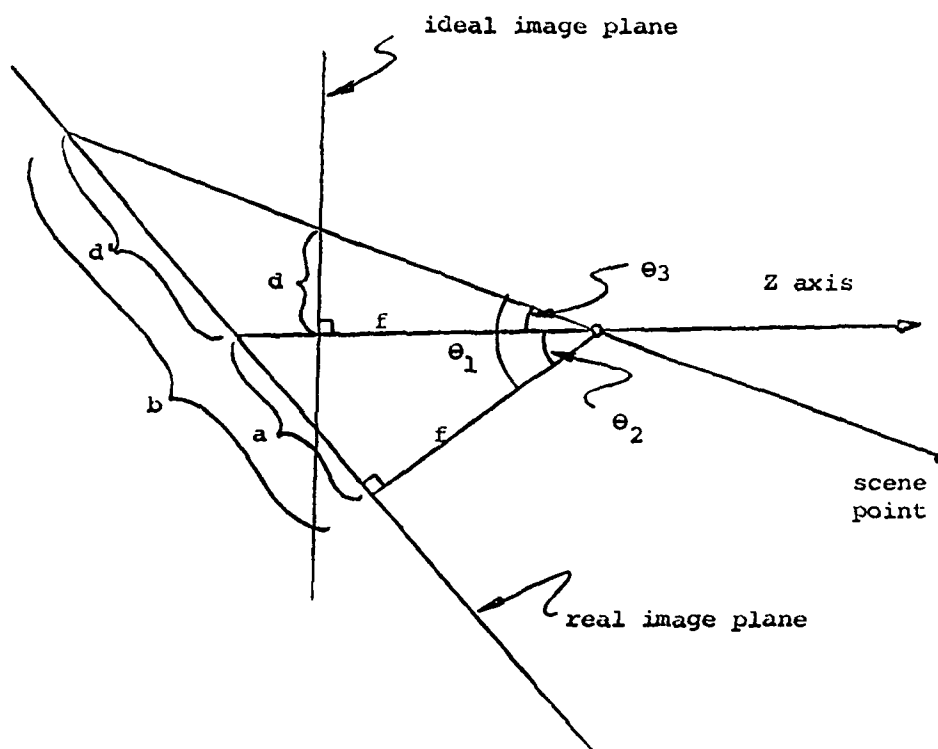


Figure 28. The relation between the intersection of a ray from a scene point falling on a real image plane and an ideal image plane

The center of the real image and the position of the FOE relate the principal camera ray to the Z axis, effectively defining the pan and tilt angles. To simplify, in our diagram only one angle is considered as shown in figure 28. It is obvious that  $\theta_3$  is the difference between angles  $\theta_2$  and  $\theta_1$ , and the focal length times the tangent of  $\theta_3$  gives the position on the ideal image plane. Angles  $\theta_2$  and  $\theta_3$  are easily computed using inverse tangents functions. The resulting ideal position of a real camera point, relative to the FOE is given by the expression:

$$x_i = f \tan \left( \tan^{-1} \frac{x_r}{f} - \tan^{-1} \frac{x_c}{f} \right),$$

(6)

where  $i$  is used as a subscript for the ideal image point,  $r$  is used as a subscript for the real image point,  $c$  is used as the subscript for the principal camera ray or optical center of the real image, and  $f$  is the focal length of the camera. All  $x$  values are relative to the centers of the corresponding images. This difference between the position of the image point in the ideal and real images increases near the edges of the image, and increases as a function of the difference between the orientation of the real and ideal image planes. For long

focal lengths and small film planes this error function is very similar to a hyperbolic, as discussed in (Prager 1979).

This difference can be plotted as the change in position (difference in number of pixels) arising from a given change in FOE position (again in pixels), and the position of given image points. Actual error functions were derived for the camera and experimental data that were used, and are presented in chapter IV.

The  $t_0$  and  $t_1$  images are registered so that points far away, especially those near the FOE, line up when one image is laid upon the other. This registration was done interactively. It is accomplished by a translation and a rotation in the xy plane.

The registration values (x and y translation and rotation) could be used via relation (6) to then warp one image so that it would appear exactly as if the camera had not rotated, panned or tilted between frames. This was not done however, since registration compensated for the majority of the effect (about 98% of it in the worst case), with a residual error of less than one tenth of a pixel (see chapter IV) in the worst case. Also, compensation of the residual error would be require

computation of tangent terms for every image point would produce a very heavy computational burden.

III.1.5 Surface orientation. Surfaces are the outer faces of objects, and delineate each object from its environment. The surface interpretation process deals with visible outer surfaces as its basic entity. A set of contiguous pixels of an image is called a region of the image, and contiguous visible portions of surfaces that are homogeneous can be segmented in the image as regions. Unfortunately, there does not exist a one-to-one correspondence between segmentable regions and surfaces because of factors that have been discussed (see chapter II). The use of resegmentation (as discussed in section III.2.6) is an attempt to correct for this non-correspondence.

The relation between the motion of image points and distance to the corresponding scene points has been specified (5). The motion of image points hypothesized to be on scene surfaces can now be examined. For instance, a surface parallel to the image plane will have the same value of  $Z$  for all points on that surface.

The surface interpretation is based on an assumption that all scene surfaces are planar and are in one of two orientations: 1) a constant Z plane, i.e., parallel to the image plane, or 2) a constant Y plane, e.g., the road surface (see figure 29). We will not consider here the planar surface of constant X, which is perpendicular to the road surface, although the relations are very similar to case 2 above.

Surfaces in the real world might not satisfy these assumptions of planarity and orientation. Real surfaces are curved, and found in an infinite variety of orientations. Although planar surfaces do not accurately reflect the nature of real surfaces, they are sufficient for recognizing objects at a distance. Consider a telephone pole, or a tree represented as rectangular solids. The dimensions of height, width, and depth of the real curved object are preserved reasonably well. More detailed descriptions of surface orientation would be needed if it were necessary to distinguish between objects of the same size but different shape, or if it were necessary to determine details of their three-dimensional shape.

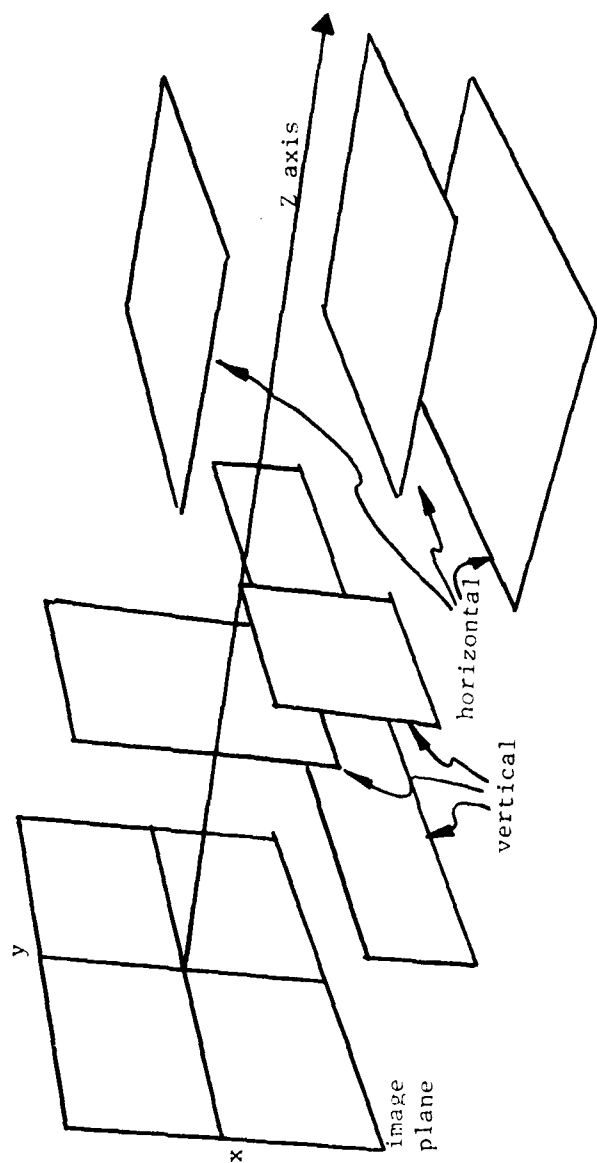


Figure 29. Examples of the two orientations of surfaces which this thesis uses in its representations

Adding the third orthogonal planar orientation of  $YZ$ , i.e., constant  $X$ , is certainly possible, and would complete the trihedron of planes. The inclusion of this third orientation is left for further work, in applications where its specification is needed. No surfaces appear in our example scene that would require its use. It is a rather straight-forward extension of the techniques presented here with the major problem being the selection of the three orientations as a best choice for the surface of unknown orientation. Use of orientation other than a small set of fixed ones would be an extension of this work, increasing the search problem.

We call a surface of constant  $Z$  a vertical surface since such a surface is parallel to the image plane. Thus, the  $Z$  value in equation (5) is the same for each point  $P$  believed to lie on any given vertical surface.

We refer to a surface of constant  $Y$  as a horizontal surface because, for our images, such surfaces correspond to the horizontal surfaces in the scene (in general, they will be parallel to the ground). Let us examine this case in a bit more detail.



For a particular image point representing the projection of a scene point lying in a horizontal planar surface, the value of  $Z$  depends on the value of  $Y$  (the height of the surface relative to the  $Z$  axis), on  $y$  (its height in the image), and on  $f$  (the focal length of the camera). Consider the simplified drawing of figure 30. Because our formulation of image dynamics is cast in terms of  $Z$  (the distance to a scene point), first we determine the distance to a scene point on a horizontal surface in terms of its height, and then substitute in equation (4).

By similar triangles (in the simplified drawing figure 30) we observe that:

$$\frac{f}{y} = \frac{Z}{Y} \quad \text{or,} \quad Z = \frac{f Y}{y} \quad (7).$$

Substituting this value for  $Z$  in equation (4):

$$\Delta y = \frac{\Delta Z y^2}{f Y}, \quad \Delta x = \frac{\Delta Z x y}{f Y}. \quad (8)$$

III.1.6 Features chosen. The motion projection relations (4 and 7) will be used to produce hypothesized displacements for inter-image comparisons. These comparisons are carried out by differencing point feature values from the two images (according to predicted

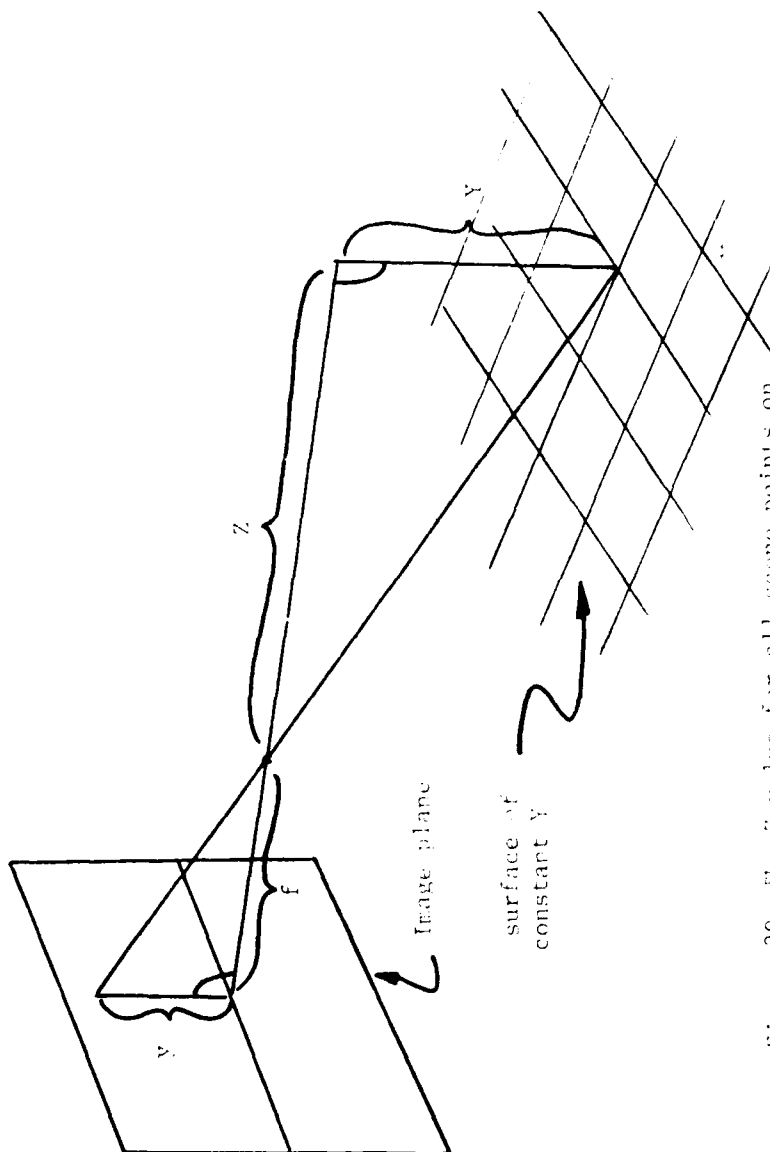


Figure 30. The  $Z$  value for all scene points on a horizontal surface is a function of  $Y$  or  $y$  and  $f$  as shown by these similar triangles.

displacements). Differences are used as an error measure to refine the model from which the displacements were derived. Additionally, the initial model is derived from point feature aggregations. We choose one point feature to accomplish both inter-image comparisons and region segmentation for the initial model. The algorithm explained below generates a feature that is sensitive to the distributions of data within the images. It was developed by (Nagin 1979) for use on static scenes. This feature is constructed from one or more initial features.

First, an initial feature, such as color or intensity is selected by hand. The feature value for each image point is calculated, and a histogram of all the point values is generated. Image data often form clusters in the histogram that correspond to populations from visually distinct areas. Each cluster  $i$  is given a unique label,  $\lambda_i$  (see figure 31), and a vector of distances in feature space from each point to all cluster centers is computed. We make the assumption that the likelihood that a pixel value is a member of a cluster is inversely proportional to the distance between the point's position and the cluster's center in feature space. The inverse of each distance element is computed and the vector is normalized so that its elements sum to

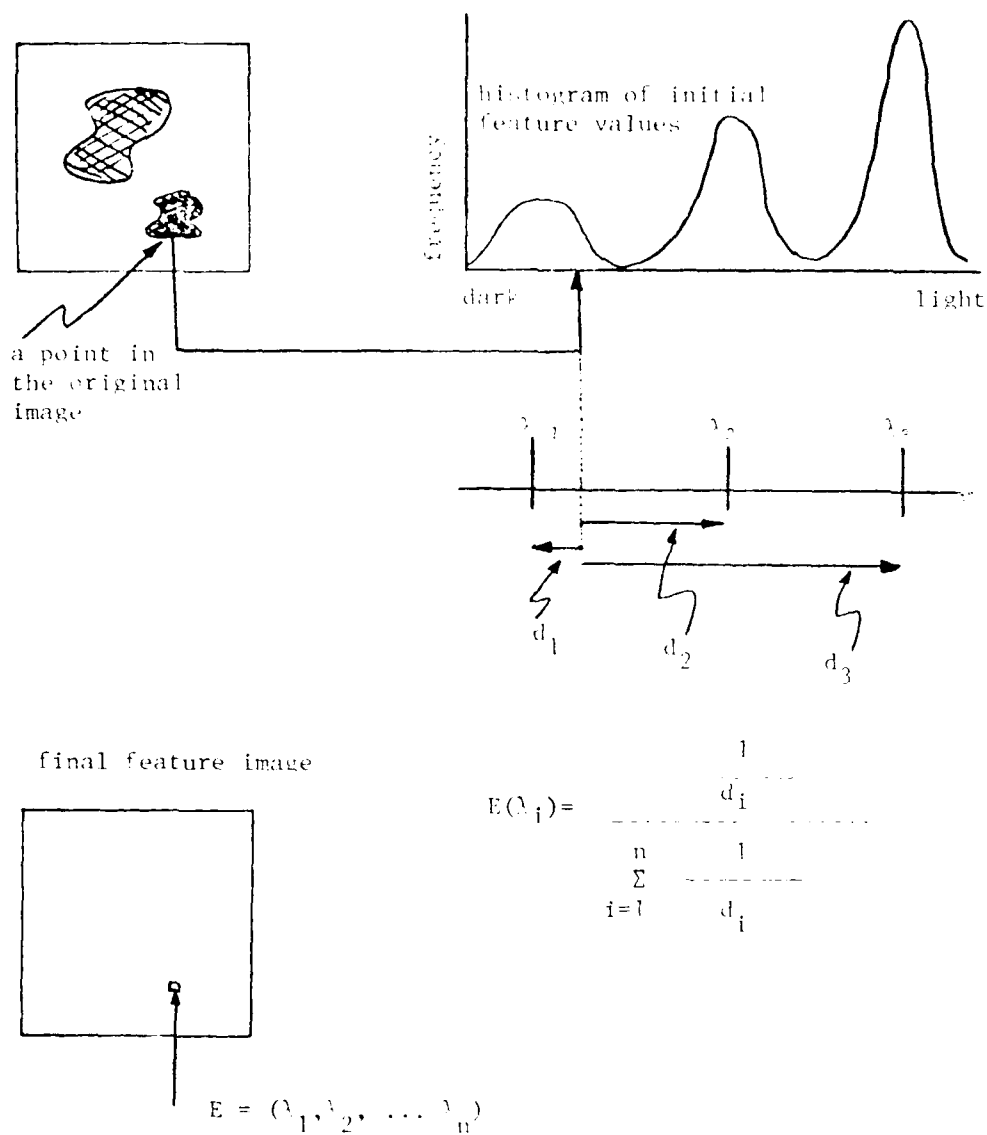


Figure 31. Simplified diagram of feature derivation. Distances in an initial feature space from a point to cluster centers are used to obtain the final features.

one. These vectors (one for each image point) are the features that we use in our feature image both for initial segmentation, and for inter-image comparison.

Clearly, this general process is extendable to any number of features, using a multi-dimensional histogram that could combine color and intensity features. The single feature example was chosen for clarity. Refer to (Nagin 1979) for more details on this algorithm and for improvements in its performance which can be obtained by localization of the area over which histograms are computed. Automatic selection of good features for this system is being investigated by Kohler (Kohler 1980).

The point feature values (vectors) are used for interframe matching in the refinement process of our surface interpretation. Point features are not the result of interpixel differencing, and therefore do not fall prey to the problems associated with edge or vertex features (see chapter II). Point features can be aggregated in a variety of ways to produce segmentations depicting regions of similarity in color or intensity. These regions are used to delineate the hypothesized surfaces in the initial surface model.

III.1.7 The initial model. Before model refinement can take place, an initial model must be supplied. We supply our system with an initial model that contains as little human-supplied information as possible. When the system has no model of the visible environment, an initial segmentation is derived using the technique described in the previous section. This region analysis is performed by aggregating the point features (section III.1.6) into regions. Each resulting region is initially assumed to be a single distinct surface. In order to clearly test the robustness of the refinement process, a number of initial Y and Z assignments are made.

Note that any reference to a surface in the surface interpretation model is a reference to a hypothesized surface. Each hypothesized surface has a corresponding region in the image. The regions are either derived from the initial segmentation process (if the system were starting anew), or could be composed of regions resulting from the resegmentation and/or remerging process described in the next section of this chapter. Thus, the term "region" refers in general to an aggregate of pixels based upon an analysis of features from a number of sources, although a region is constructed from the image data alone when constructing a new model.

The initial segmentation is automatically derived by aggregating the feature image at time  $t_0$ . This aggregation is performed in four steps (see figure 32). The first step is simply the generation of the feature image, where the value of each pixel is a vector of normalized inverses of distances to cluster centers in the original feature space.

In the second step the vector for each point feature is examined to find the label with the maximum likelihood value. A new image is created with label numbers for each pixel. These labels correspond to the nearest cluster center for each point. The maximum label image is effectively a segmentation based on a minimum distance classifier, where the target set consists of cluster centers.

In the third step, a plurality update rule is applied to  $3 \times 3$  windows of the label image. This rule is an update function for the label of the central point (Nagin 1979). This label is replaced by the label associated with the mode of the labels in the window surrounding the point (see appendix). The plurality update rule allows efficient computation at the expense of yielding a cruder segmentation than that available

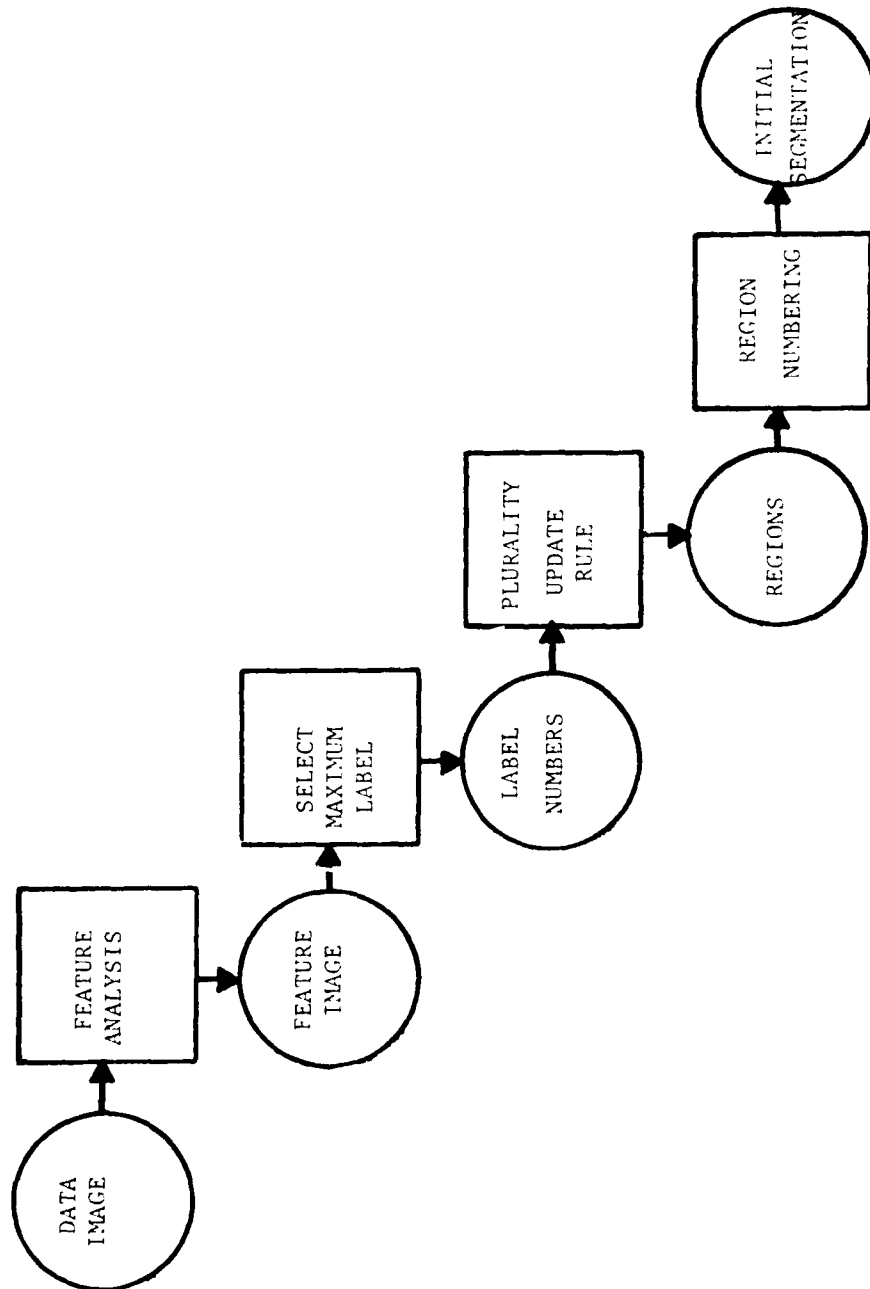


Figure 32. The initial segmentation is produced in four steps



from other update rules. In the experiments of chapter IV we use two iterations of this update rule. This rough segmentation is used as an initial plan for further processing and refinement.

The fourth step assigns a unique region number to each contiguous set of pixels with identical labels. Thus, an initial segmentation is obtained.

The initial model consists of an image segmentation where each region is assumed to be a separate surface. Each region is given a label  $j$ , and initial values  $Z_j$  and  $Y_j$ . The initial  $Z$  and  $Y$  values for all regions are set to the same value by the user of the system. The effect of choosing different initial uniform values is described in the experiments in chapter IV.

### III.2 The Surface Interpretation Process

The primary functions of the surface interpretation process are the discovery and/or refinement of distances to vertical surfaces and heights of horizontal surfaces in the scene. A secondary, but necessary function is the discovery and refinement of the Focus Of Expansion. These functions are carried out by a search process that synthesizes images according to predictions of the

motions of image points.

Figure 33 shows an example of correctly predicted displacements for two hypothetical regions and several points. Figure 34 shows the effect of changing the FOE, while figure 35 shows the effect of changing the Z values for each surface.

Predicted image motion is based upon the currently hypothesized surface interpretation. Thus, an hypothesized model or FOE value is tested on the basis of detected differences between it and displacements in the image pair. The surface model is accurate if it accounts for the displacement of image points between movie frames. Refinement refers to a search process which utilizes successive hypothesis-test steps.

Recall that the surface interpretation process consists of three major phases (see figure 36 which is similar to figure 22 from the first part of this chapter). The first is model initialization. For this, an initial segmentation is derived as described above in section III.1.7, and each region is considered as a distinct surface. Then, this set of surfaces is assigned initial surface distance and height values (as described in chapter IV). This completes the model initialization

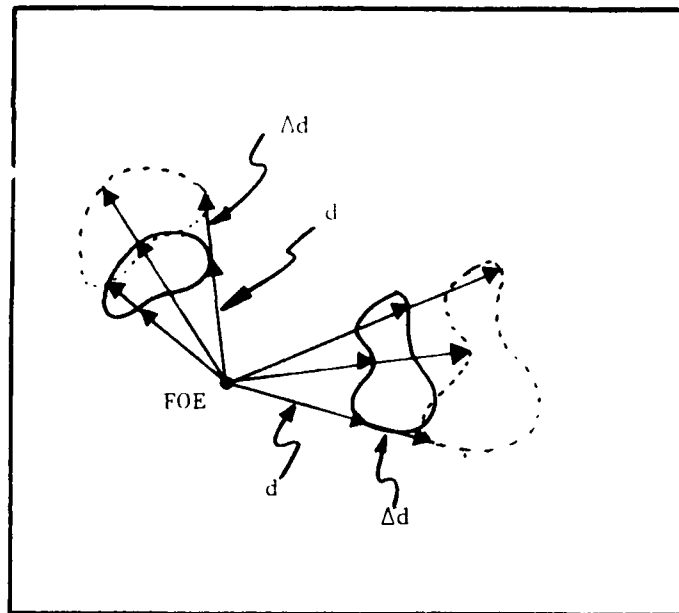


Figure 33. Relationship between FOE and displacement vectors

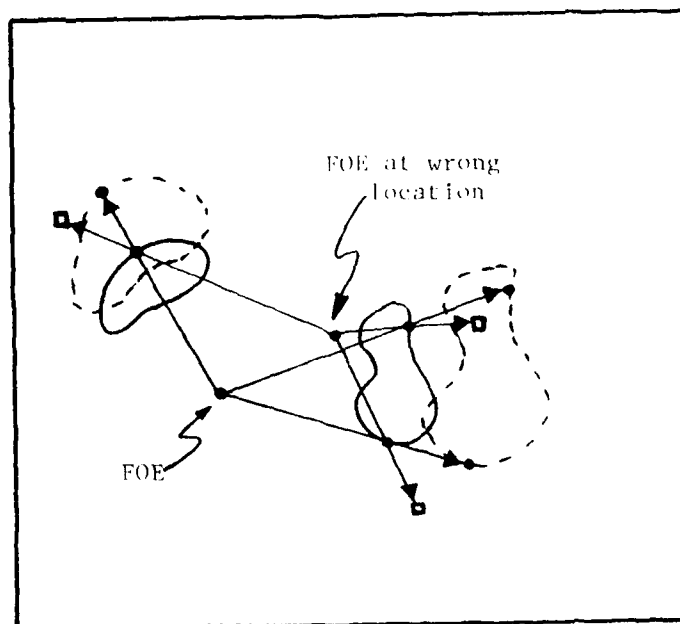


Figure 34. Effect of placing FOE at incorrect location is to change the direction and length of displacement vectors as indicated by the small boxes in this figure.

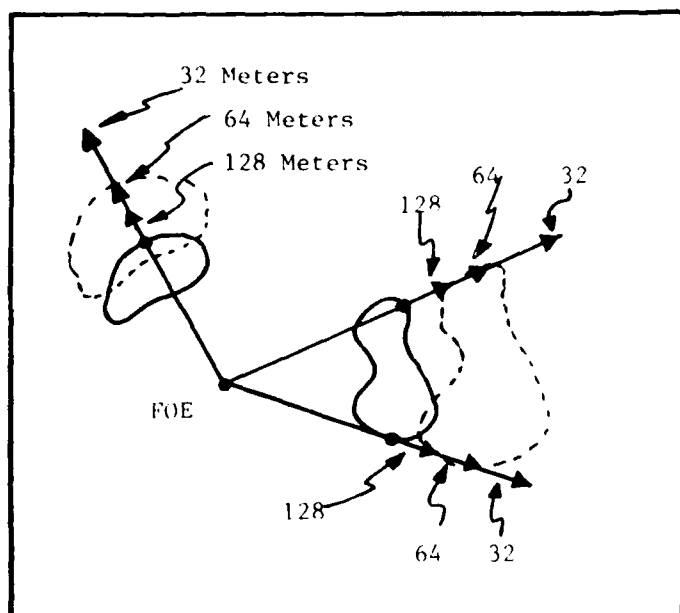


Figure 35. Effect of changing the Z values for the surfaces is to shorten or lengthen the displacement vectors. In this figure the correct distance values are 64 meters.

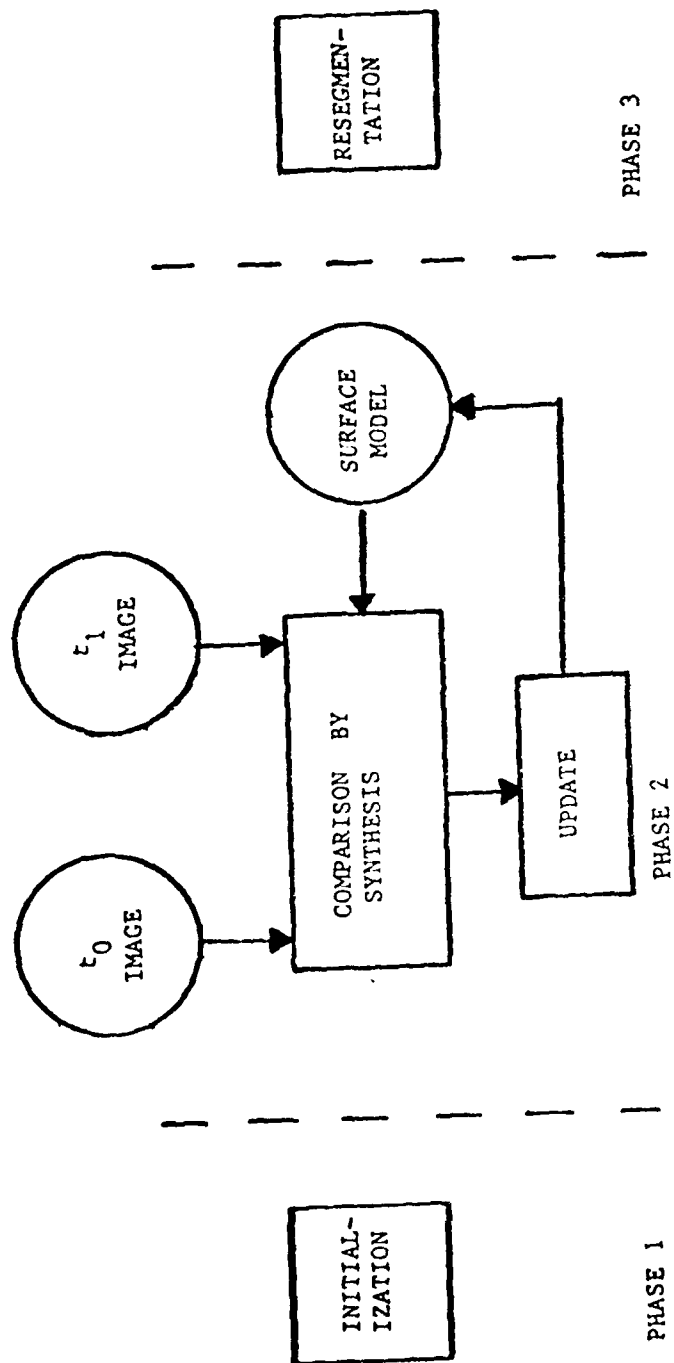


Figure 36. The surface refinement phase of the surface interpretation system is explained in this section

phase. The second phase is FOE, distance, and height refinement. This phase is central to the entire process and will receive the most detailed description in this and subsequent chapters. The final phase is a surface re-segmentation that both divides and joins hypothesized surfaces based on their behavior with respect to the best (refined) model. The resegmentation process is described at the end of this chapter, in section III.2.7.

The following discussions will contain descriptions of algorithms, the majority of which are assumed to be parallel in nature. The parallelism exists at the point level, that is, the algorithms are to be applied at each point of the  $t_0$  image or model.

III.2.1 Image synthesis. To accomplish model and FOE refinements, the system compares two feature images. One is the actual feature image taken at time  $t_0$  and the other is a synthetic time  $t_0$  image. This synthetic image is produced by warping the real  $t_1$  feature image, according to the surface model, so that it will look like the real  $t_0$  feature image if the surface model is correct.

Notice that this synthesis can be viewed as "projecting backwards" in time (from the future to the present). The choice of projecting backwards - rather than forward - is the result of choosing the  $t_0$  image as the one from which the initial model is generated. However, the computational mechanism is basically the same in either case.

Image synthesis is accomplished through a process which checks for occlusion, computes the displacement for each point, and interpolates the  $t_1$  feature image for sub-pixel resolution. We wish to compare a discrete pixel value in one image to a corresponding pixel value in the second image. The pixel value in the second image is obtained by interpolating a synthetic value from a window of feature values around the tip of the displacement vector. The synthesis system is diagrammed in figure 37. The occlusion and interpolation subsystems are described below.

III.2.1.1 Occlusion. In our images, changes in occlusion are the result of camera motion. The nearer of two objects has a smaller value of  $Z$  than the farther object. Equation (4) implies that smaller  $Z$  yields larger image displacements ( $\Delta x$ ,  $\Delta y$ ), and therefore



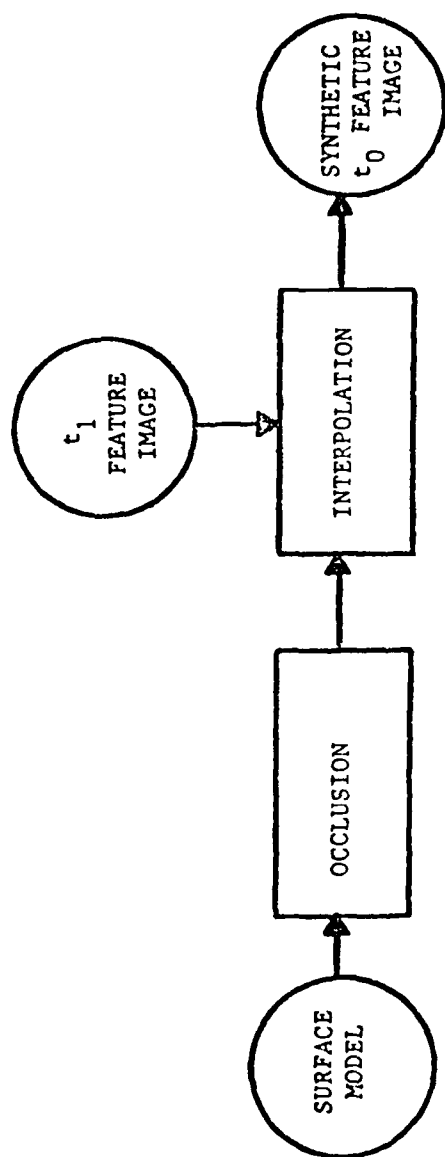


Figure 37. The production of the  $t_0$  synthetic feature image is explained in two stages

imaged points from an occluding object must be moving faster in the image than those from an occluded object. Figures 38 and 39 show this effect and identify the areas that are about to be occluded and "disoccluded" in the successive frames. If the system does not take into account the areas of occlusion, significant errors in the comparison process would result. Therefore, the system uses the hypothesized surface model to ignore, in the matching process, those areas that are occluded and disoccluded between two frames.

Unfortunately, it is not feasible (in general) to determine occlusion at the same time as the inter-image comparison; instead a temporary  $t_1$  surface model is generated. Consider the problem of determining if a point in the  $t_0$  image is going to be occluded in the  $t_1$  image. To predict this one has to know if any closer surface will move to the predicted location of that point in the  $t_1$  image. This cannot be known unless it can be determined that every point of every surface (between the predicted location and the FOE) will or will not occupy that predicted location. Thus, prediction of occlusion is a fairly global problem, and the locality can only be constrained further by restricting the velocity of image points. Furthermore,

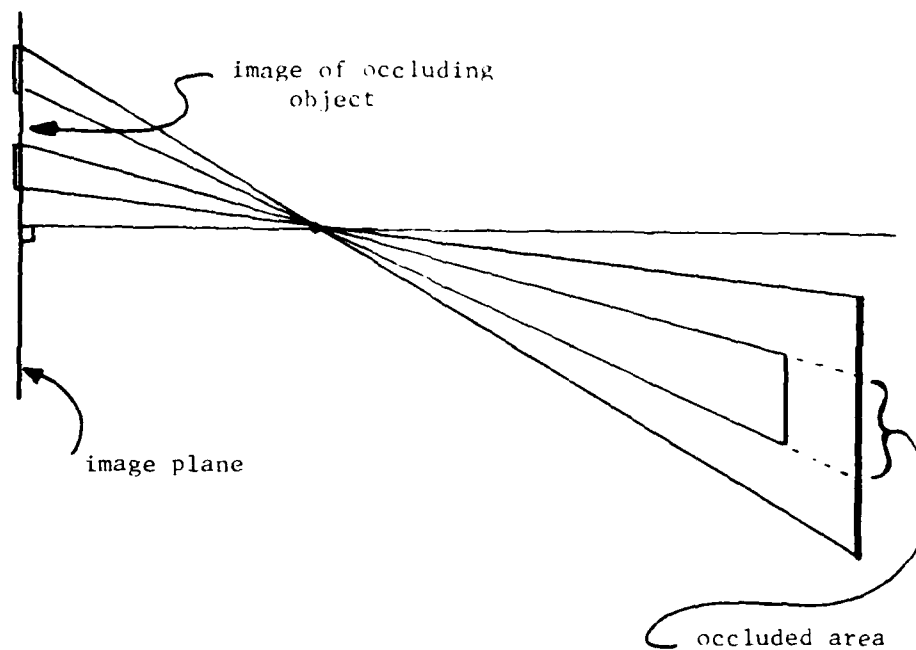


Figure 38. The projection geometry determines the effect of occlusion.

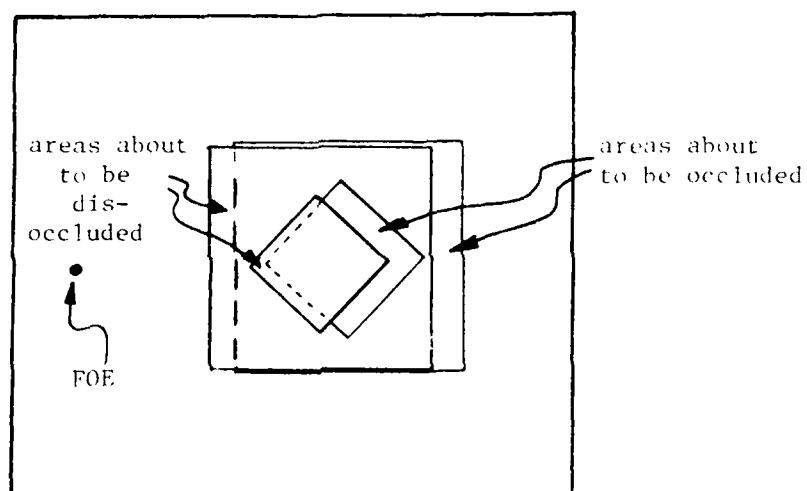


Figure 39. Motion of the camera causes occlusion and disocclusion.

the amount of local computation required, if such a scheme were employed, would be considerably greater than the amount required by the single-pass algorithm presented below.

The first step performed in the prediction of occlusion is the formation of a temporary  $t_1$  surface model (see figure 40). prior to inter-image comparison. Each surface of the  $t_0$  model has an associated value for  $Z$  and  $Y$  that (through equations 4 and 8) yields a predicted displacement (in the image) for each point of that surface. These displacements are used to direct the generation of a  $t_1$  surface model.

Many different visible points in the  $t_0$  model which are at different distances could be predicted to appear in the same location in the  $t_1$  model. The algorithm employed predicts the displacement of each pixel based upon  $Z$  and  $Y$  values, and then selects the minimum at each location. Since the algorithm is implemented as a sequential process, that is, each point is visited only once in the  $t_0$  model, then the algorithm will start with a  $t_1$  model which is initialized with 1000 meters. Then the process checks the value at the predicted location in the  $t_1$  model,

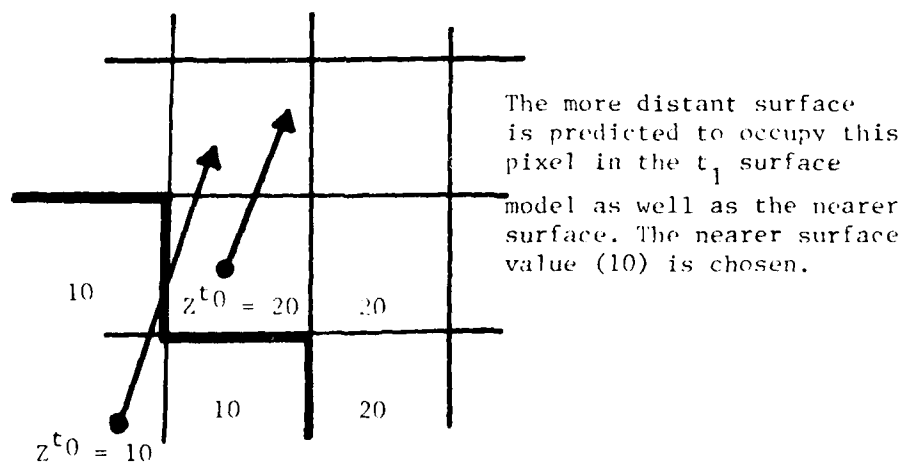


Figure 40. An example of a surface at 10 meters and another at 20 meters competing to assign a value to the  $t_1$  surface model

compares it with the  $t_0$  value and deposits the lower of the two.

Two methods are used for generating the temporary  $t_1$  surface model, but can both be expressed as one algorithm (see figure 41). One method is used in preparation for Z and Y refinement and the other is used in preparation for FOE refinement. Recall that the model consists of a Z value and a Y value for each surface.

In the case of Z and Y search the Z and Y values are separately projected to the  $t_1$  model. Thus, the Z value for each surface in the  $t_1$  model is computed by projection of the Z values of the surfaces in the  $t_0$  model, and the Y values for the  $t_1$  model are computed by projection of the Y values of the surfaces in the  $t_0$  model.

In the case of the FOE refinement, another method is employed for generating the temporary  $t_1$  surface model. A choice is made whether each surface is to be vertically or horizontally oriented. In this case the  $t_1$  model is a special type of surface model. It is one with either a separate Z or Y value for each surface, but not both. The choice is based on the error measure which is produced during the last (previous) search step of the

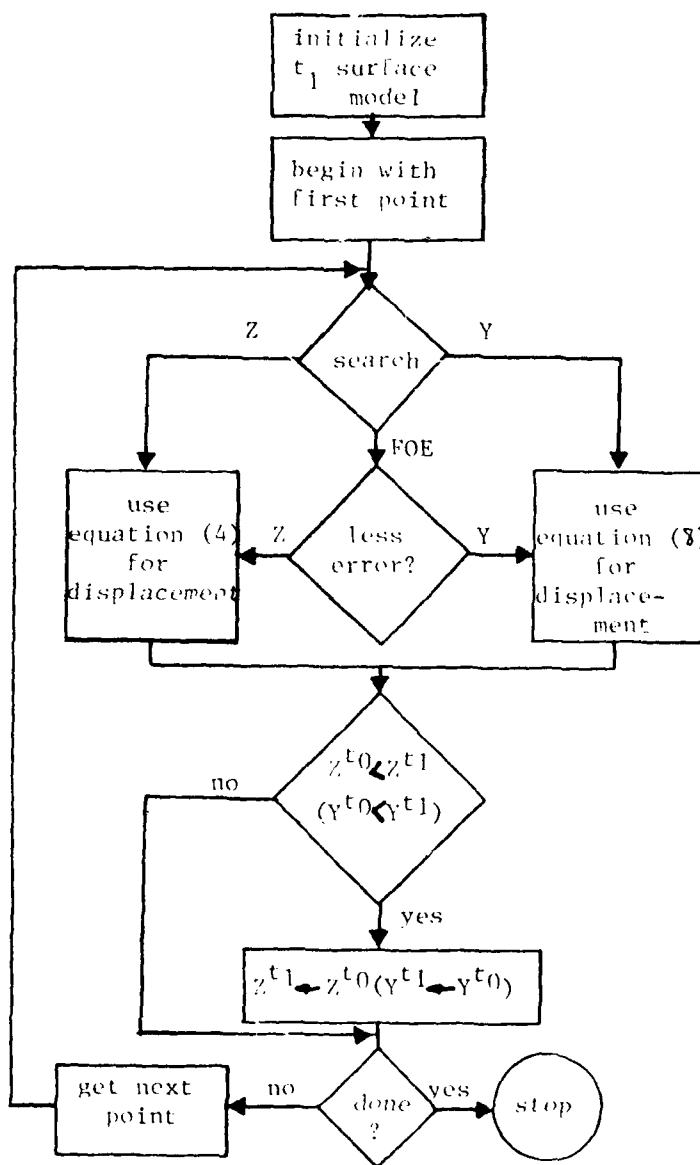


Figure 41. Flow chart of algorithm that produces the  $t_1$  surface model



refinement process. The error measure is explained below, in section III.2.2. Notice that the  $t_1$  surface model is only used to predict occlusion. By making the Z vs. Y choice, the "best-guess" surface model is used to predict occlusion during FOE refinement.

III.2.1.2 Interpolation. The synthetic  $t_0$  image is generated from the  $t_1$  image in basically three steps (see figure 42). First, the displacement vector for each  $t_0$  point is computed from equations (4) and (8). The base of each displacement vector rests at the center of a pixel, and is represented by a pair of integers. The tip of the vector however, is dependent on the projection relations, and typically is not positioned at the center of a pixel.

The second step consists of a test prior to interpolation. Consider the activity relating to one point in the  $t_0$  image. The displacement vector is derived from the equations (4) or (8). The choice of which equation to use depends on whether the search is for Z, Y or FOE.

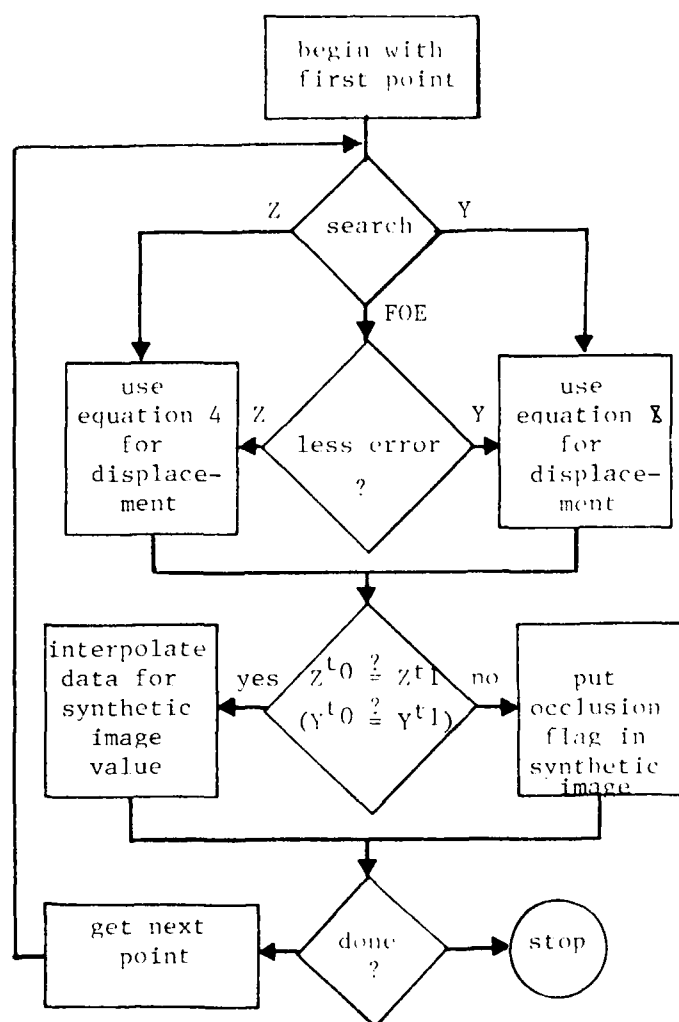


Figure 42. Flow chart of algorithm that interpolates to generate synthetic  $t_0$  image

If the point at the tip of the displacement vector in the  $t_1$  surface model contains a distance value different from the point at the base of the displacement vector in the  $t_0$  model, occlusion is hypothesized for the point. This is equivalent to discovering that this particular point is hypothesized to not be visible in the next image. The point in the synthetic  $t_0$  image is assigned an occlusion flag if occlusion is hypothesized. This flag prevents the occluded points from being counted when the error values are averaged across the entire surface.

In the third step, each  $t_0$  synthetic image point that is not going to be occluded is assigned a feature value. A value is synthesized for each pixel by examining the area around the tip of the displacement vector in the  $t_1$  image. To achieve sub-pixel displacement resolution, the  $t_1$  image is to be interpolated using a bi-linear interpolation scheme. The contribution of each pixel covered by a displaced pixel-sized square is weighted according to its area (figure 43). The weighted sum forms the interpolated value for the point in the synthetic  $t_0$  image.

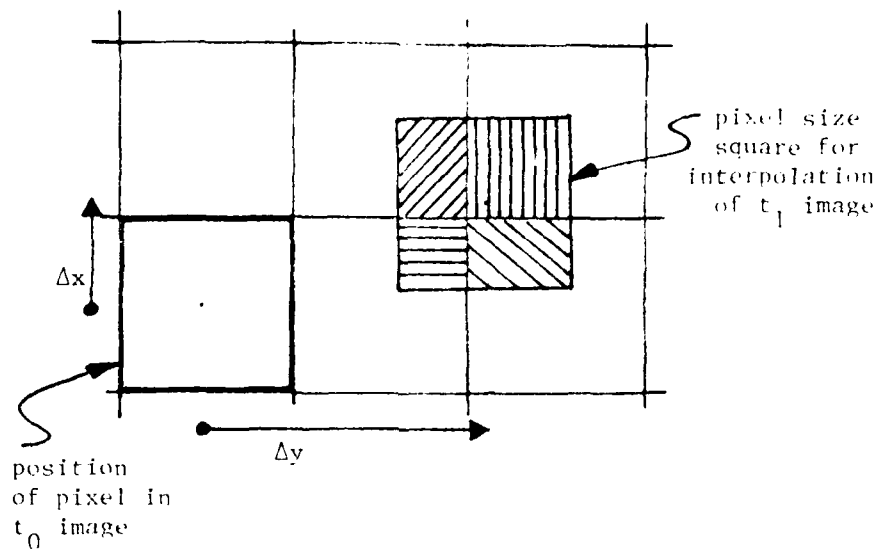


Figure 43. Interpolation is performed by taking the weighted average of the values under the displaced pixel in the second image. The weight for each value is the area covered, as indicated by the hatching in this figure.

III.2.2 Search using error images. The system employs a search technique to refine the distance and height values for each surface in the model, as well as for the FOE for the entire image. As with all search techniques, a measure is needed which indicates which of the trials is closer to the goal. The assumption made is that errors in the model will produce incorrect displacement predictions between images, and thus, would lead to large differences between actual and synthetic images. The image produced by the pointwise difference between the synthetic and the real feature image at time  $t_0$  is called an error image.

The difference values in the error image are averaged across each surface to produce an error value for that surface, and are averaged across the entire image to produce an error value for an hypothesized FOE. It is these error values that the search process uses to select among alternatives (as described below). While averaging values, the system ignores points that contain occlusion flag values, thereby ignoring areas that are predicted to be occluded in the next image.

The difference measured between corresponding points of the two images is the Euclidean distance between the two feature value vectors. We have already specified that these vectors are normalized (based on relative distances to cluster centers in feature histograms) so the sum of the elements of each vector is one. Therefore, the maximum distance between any pair of feature value vectors is the square root of  $L-1$ , where  $L$  is the number of labels (length of the vector). In order to scale all difference measures into the same range, regardless of the number of cluster labels employed in the initial segmentations, the Euclidean distance between point feature values is divided by the square root of  $L-1$ .

Now let us consider how error images will be used in a search process. We assume that incorrect FOE, Z or Y values will lead to many incorrect predictions of point displacements, which will thereby cause large values in the resulting error images. Several error (difference) images can be computed by differencing the real feature image with several synthetic images, where each synthetic image is the result of a systematic variation in the location of the FOE, in the Z value for a surface, or in the Y value for a surface (See figure 44). The image

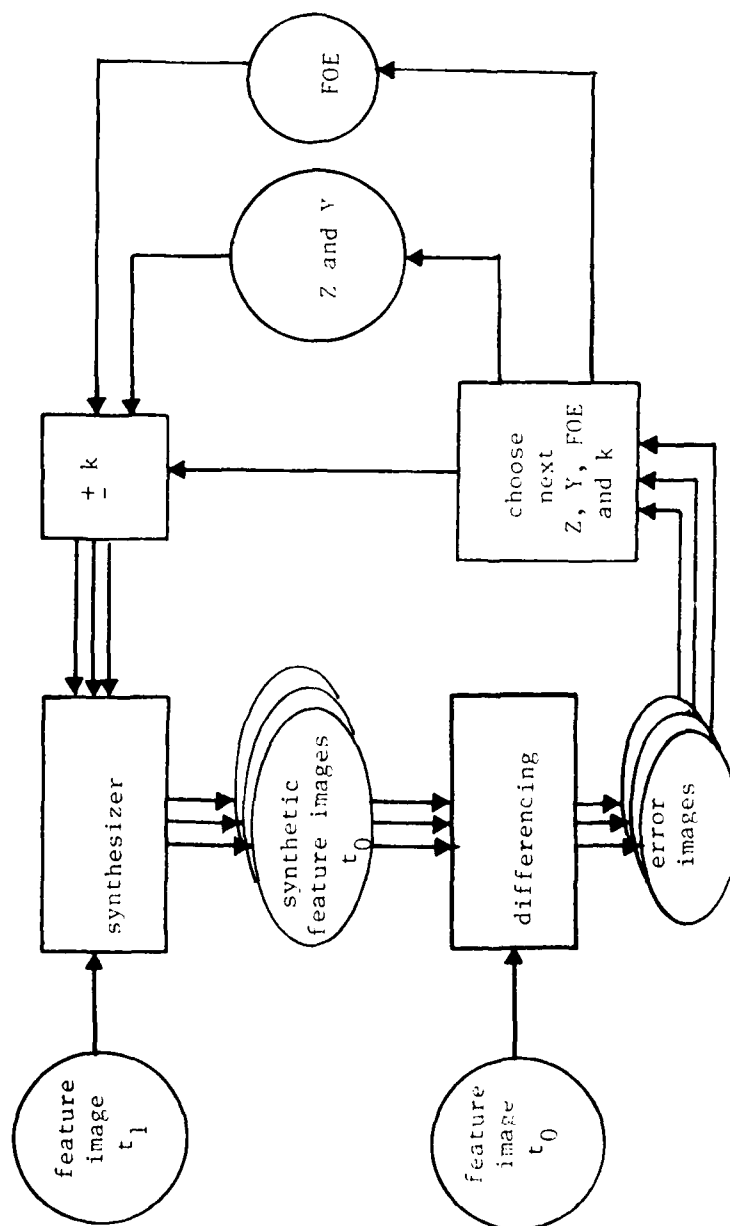


Figure 44. The search process uses error images to direct its next choices.

with the least error indicates the model which best predicts pixel movement.

The systematic variation of FOE will simply involve increments and decrements of the x and y values for the FOE. The systematic variation of Z and Y will involve increments and decrements of the values for Z and Y for each surface. These variations are used to sample an interval of the unknown error function for each value being refined. The search pattern of samples includes one central value (no change), one incremented value, and one decremented value for each value being refined.

Thus, a simple "hill-climbing" technique minimizes an error value (perhaps it should be called "valley-descending"). In general, this technique is vulnerable to getting stuck on a local minimum, although in practice we have not experienced such events. Even in such cases, since new data arrives frame by frame, one would expect that the system would recover from local minima fairly quickly.

The search is iterated, each time using the last FOE (Z or Y) value(s) that produced the lowest error measure. The sizes of the increments and decrements are reduced during the search process so that it may converge. Each



with the least error indicates the model which best predicts pixel movement.

The systematic variation of FOE will simply involve increments and decrements of the x and y values for the FOE. The systematic variation of Z and Y will involve increments and decrements of the values for Z and Y for each surface. These variations are used to sample an interval of the unknown error function for each value being refined. The search pattern of samples includes one central value (no change), one incremented value, and one decremented value for each value being refined.

Thus, a simple "hill-climbing" technique minimizes an error value (perhaps it should be called "valley-descending"). In general, this technique is vulnerable to getting stuck on a local minimum, although in practice we have not experienced such events. Even in such cases, since new data arrives frame by frame, one would expect that the system would recover from local minima fairly quickly.

The search is iterated, each time using the last FOE (Z or Y) value(s) that produced the lowest error measure. The sizes of the increments and decrements are reduced during the search process so that it may converge. Each

value is refined under the assumption that the search proceeds toward a global minimum error. Figure 45 shows an error function and several successive samples. We use the parameter "k" for the sampling increment as described in each search process below.

III.2.3 Search for FOE. Two search systems for FOE are discussed. During start-up an accurate surface model is not available. Therefore, to handle the start-up problem, an enhanced search process is used; it is presented below in section III.2.5, after the basic FOE, Z and Y searches are explained. First, however, we examine the basic FOE search process, which is effective when an accurate surface model is available.

The basic FOE search mechanism examines nine foci of expansion (see figure 46) and for each FOE it produces a synthetic feature image. These images are differenced with the actual feature image at time  $t_0$ , and the difference is averaged across the entire image, thereby producing one error value for each FOE. The FOE that results in the lowest error is chosen as the next central starting point.

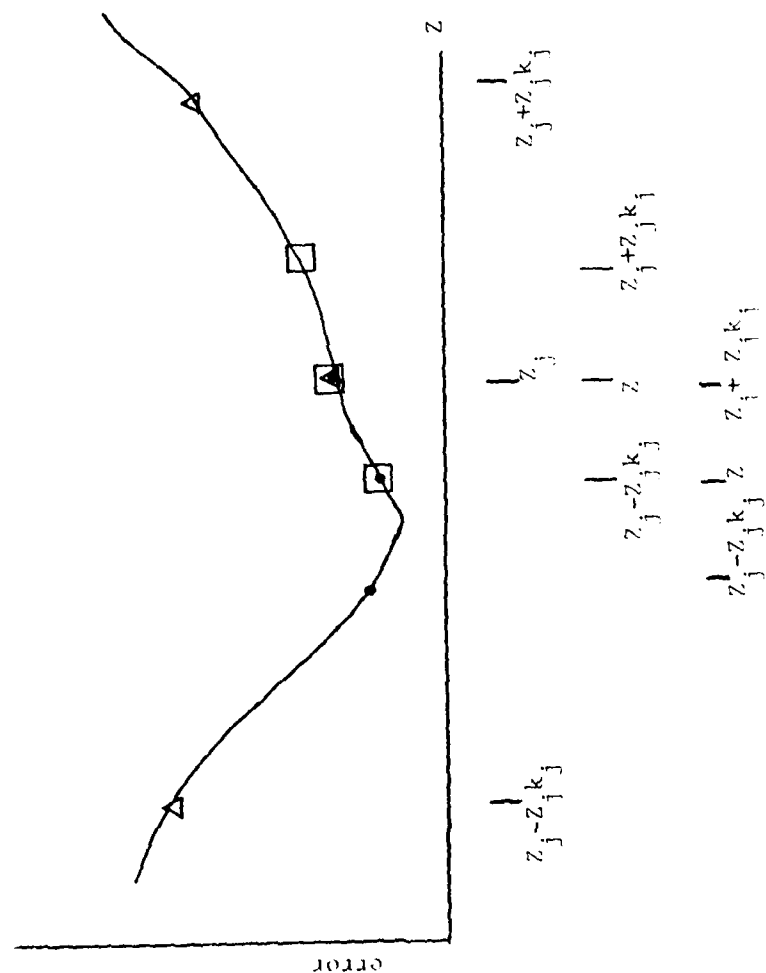


Figure 45. An unknown error function will be probed with three values for  $\gamma$ . Three such successive attempts are represented here as triangle, box and dot indicate.

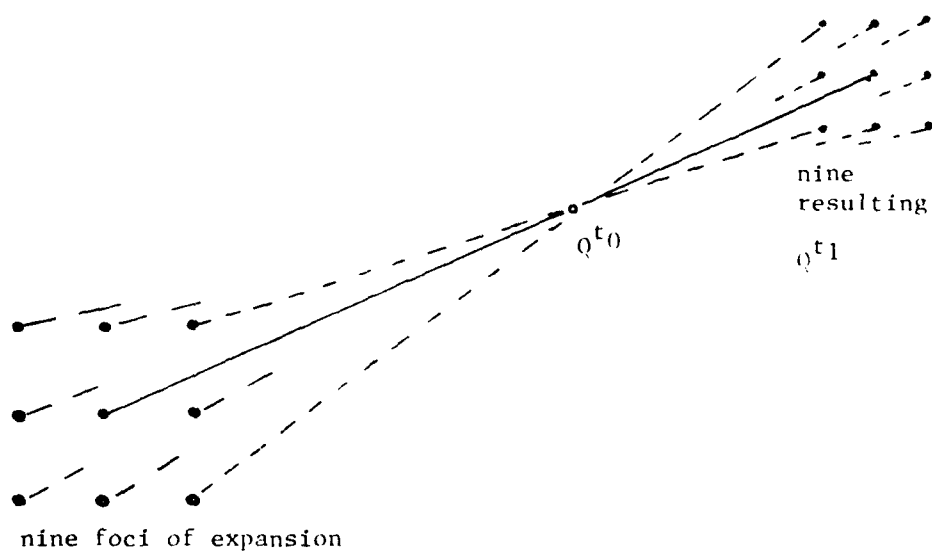


Figure 46. Nine synthetic images and their resulting error images are generated, each with the assumption of a different FOE. Here the displacements for one point are diagrammed.

An incorrect placement of the FOE would predict incorrect inter-image displacement of every point. Thus, those FOE placements that are most incorrect should produce errors of displacement resulting in large inter-image differences, or a high average error measure. Likewise, the FOE placement that is closest to correct usually should produce the lowest error measure.

The search proceeds by first selecting an initial FOE, and an initial value  $k$  for the distance increment in the  $x$  and  $y$  directions for each focus. The default starting value for the central FOE would be (256, 256) on a 512 x 512 image, and the value of  $k$  would be 128. Thus, the nine foci of expansion would equally cover the entire image. On the next search step the value of  $k$  is reduced so that the sample set is equally spread within the area between the point position that produced the minimum error value and the samples around it that did not produce the minimum. In this case,  $k$  would be 64 (See figure 47). The value of  $k$  is reduced each iteration by a constant multiplicative factor of 0.5. The FOE search selects FOE positions in discrete pixel units. It stops when it chooses between nine adjacent pixels.

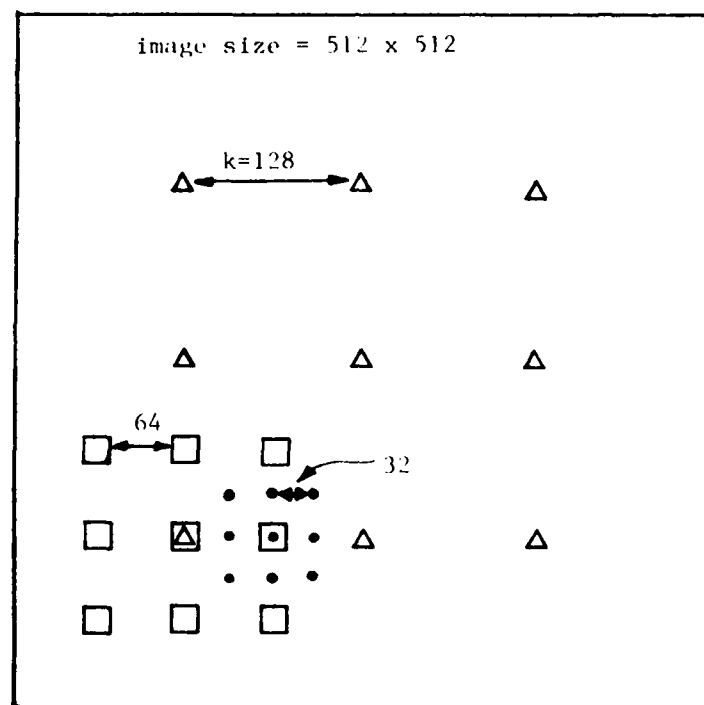


Figure 47. Successive FOF sample patterns for the first three iterations of a 512 x 512 image. The parameter  $k$  is first set to 128, then to 64 and then to 32 pixels, as indicated by triangle, box and dot respectively.

If the FOE is known to be on the image, it is possible to reduce the parameter  $k$  on each iteration. This is because the search is assumed to move toward the correct solution on each iteration and because the search space is bounded. In circumstances where the FOE is suspected to be off the image, or where assumptions of correctness for each successive search step are not valid, it would be necessary to use an appropriate initial value of  $k$ , (perhaps larger) as well as a test for reduction of  $k$ . This would allow the search to proceed a considerable distance from the image to find an FOE which, although on the image plane, is not within the bounds of the image data.

We cannot guarantee that the global minimum error will be found by only selecting nine sample foci and searching. However, our experience to date has demonstrated that this sample set and technique has been adequate to locate the FOE in our image data. This search takes  $9((\log_2 N)-1)$  full image differences to cover an image of  $N \times N$  points. This value is 56 and 72 for  $N$  equal to 128 and 512, respectively. Expressed in terms of search iterations, each consisting of the nine foci to be tried, the numbers would be 6 and 8.

#### III.2.4 Search for the Z and Y values for surfaces.

Recall that the search mechanism for distance and height of surfaces uses a simplified model of the physical environment where there are only two orientations of surfaces. Each of these orientations is parallel to a plane containing two of the axes in the coordinate system. Therefore, we need only determine one value for each assumed orientation, and choose the orientation that best accounts for image dynamics. The determination of each surface distance or surface height involves a search among values that predict the actual image dynamics. This search takes place separately for each surface.

The search for the Z and Y values begins with an initial model of values for each surface. These initial values can come from a previous surface interpretation, but here we start the system by introducing values by hand. Experiments in chapter IV examine the effect of various starting values. The values we introduce are the same for all surfaces, i.e., a uniform Z and Y for the initial model. This represents the start-up case, where there is no knowledge of the scene.



The search strategy selects three values:  $Z_j - k_j Z_j$ ,  $Z_j$ , and  $Z_j + k_j Z_j$  for each region  $j$ , and similarly for  $Y_j$ .

An initial value of  $k$  for each surface is given as 0.5. Thus, if the initial  $Z$  were 100 meters and the initial  $Y$  were 1.0 meters, the three values for each  $Z$  and  $Y$  during the first iteration of the process would be 150 meters, 100 meters, and 50 meters for  $Z$ , and 1.5 meters, 1.0 meters, and 0.5 meters for  $Y$ .

We reduce the parameter  $k$  for each region  $j$  when the current (central) value for  $Z$  or  $Y$  produces less error than either of the other two search values. When a central value is chosen, we assume that the final value will be between

$$Z_j + Z_j k_j \quad \text{and} \quad Z_j - Z_j k_j, \quad \text{and therefore}$$

decrease the  $k$  for region  $j$ . We select a multiplicative factor of  $\frac{\sqrt{2}}{2}$  since this produces a relatively gentle

reduction of  $k$  by 0.5 for every two iterations, at least in the case where the central value has the least error.

The search mechanism is applied independently for each surface and employs three stopping criteria for the Z and Y search: a) the attainment of a Z greater than 1000 meters or less than 20 meters, b) the application of 20 iterations of the search process, or c) the attainment of a k of .025.

The limits on absolute distance stop the refinement if the surface is either too distant to register any discernable image velocity, or too close to be visible in both images. When either condition occurs there is no reason to continue the search.

A "k" of .025 indicates an attempt to resolve distance to  $\pm 2.5$  percent, a figure more precise than actual scene distance measurements (see chapter IV). Given our orthonormal surface approximation, there is no point in resolving model accuracy beyond that which can be directly measured at the scene.

If a surface has not reached one of the other criteria within 20 iterations, it typically is either very close to doing so, or it is oscillating because of an incorrect initial model. This latter case, and recovery mechanisms, are discussed in section III.2.6.

The Y search is stopped if Y attains a value less than -3.0 meters, if the search proceeds 20 iterations, or if the value of .05 is attained for k. The 3.0 meter criterion is used to "catch" any attempt to go beyond actual scene surface heights (for those surfaces with horizontal orientation). In the experiments presented in the next chapter very few surfaces reached the 3.0 meter criterion. The criterion for the parameter k is twice the value as for the Z case, again reflecting our ability to compare the results with actual scene measurements.

#### III.2.5 Decoupling the Z search from the FOE search.

Consider the application of Z value refinement with the FOE grossly in error. Let us suppose that the FOE were placed to the upper right of the sub-image, while the correct position is off the sub-image to the lower left. The Z refinement would be based on displacements that are grossly incorrect - wrong displacement amplitudes and directions - everywhere in the sub-image! If the image had reasonably discernible variations in the values of spectral features, then false matches and large computed error would occur regardless of the Z chosen. This would always be true of areas of the image that are on the "wrong side" of the incorrectly placed FOE because predicted image displacements of those areas would be in

a completely erroneous direction. This conclusion was supported by an experiment (not reported upon here) that showed areas on the wrong side of the FOE to be diverging from the correct Z value. The conclusion is that the Z refinement should not proceed when the FOE is grossly in error.

Consider now the converse problem. If an initial model is grossly incorrect and we attempt to search for the FOE, will we run into the same sort of difficulty? Our conjecture was that the direction rather than the amplitude of predicted displacements would have the greater effect upon the computation of error. The conclusion, if this conjecture is sound, is that FOE search can proceed with minimum information ( a ballpark figure of uniform Z across the image), while Z refinement cannot proceed unless a reasonable FOE is used.

This conjecture has one failing. Consider an image as shown in figure 48, where the numbers are meant to represent the actual distances to the surfaces. Given an initial uniform model, where each surface is hypothesized at 64 meters, the FOE search will be driven toward point "b" rather than the correct point "a" in figure 48. This is because, at point "b" the predicted displacement

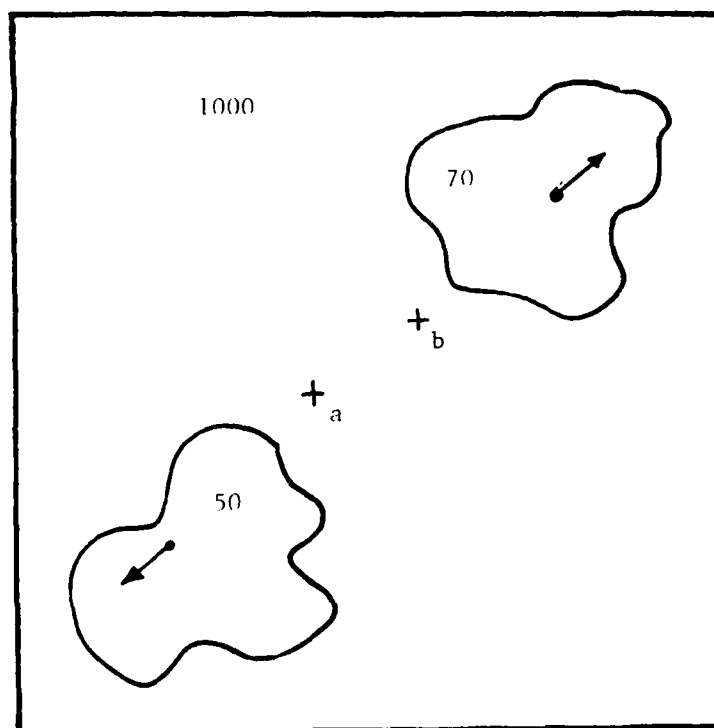


Figure 48. The locus of expansion can be incorrectly chosen if there are surfaces on either side of the proposed foci which are greater and less than the hypothesized  $Z$  values.

amplitudes for both surfaces are closer to the actual displacement amplitudes.

The points on the surface which is actually 70 meters away will have average displacements of 2.0 pixels amplitude in our example. With the hypothesized Z of 64 meters, the hypothesized displacements with the correct FOE will be 2.2 pixels of amplitude. If the incorrect FOE, point "b," is chosen, under the hypothesis that the surface is 64 meters Z, the average displacement of 2.0 pixels amplitude will result. Because this amplitude is correct (for the data) and the displacement direction is nearly correct, the error will be minimized with the selection of point "b" for the FOE. Similarly, the surface with a Z of 50 meters, under the hypothesis of a Z of 64 meters, will drive the FOE search to point "b".

Thus, asymmetries in the spatial distribution of surface distances will have an effect on the FOE search when the model is incorrect.). Furthermore, once the FOE is placed at point "b", successive Z search steps will continue to affirm 64 meters as a correct Z value for both surfaces. Then, if more FOE search steps are performed, the FOE will remain in the vicinity of point "b". The error could be unrecoverable, and could cause

similar errors in successive frames, until motion of the camera sufficiently changes image composition and dynamics.

The problem is solved by decoupling the FOE search from the errors in Z values by an enhancement we call "weighted-error". The contribution of each pixel's inter-image difference value to the average error value is weighted according to the pixel's position. This scheme makes the search more sensitive to the direction of displacement, and less sensitive to the amplitude of displacement. The weighted-error enhancement is meant to be used during the start-up phase, i.e., the first two frames of a sequence. Once a good model is obtained, the simple FOE search should be used because it is computationally more efficient.

Now we will examine the directional weighting of difference values. We will consider three foci along a line during the FOE search process. For our example we chose the lower left, central, and upper right foci (see figure 49). Displacements of points lying near or on the line that includes the three foci of our example will show the greatest variation in terms of their amplitude as a choice is made among these three foci. The

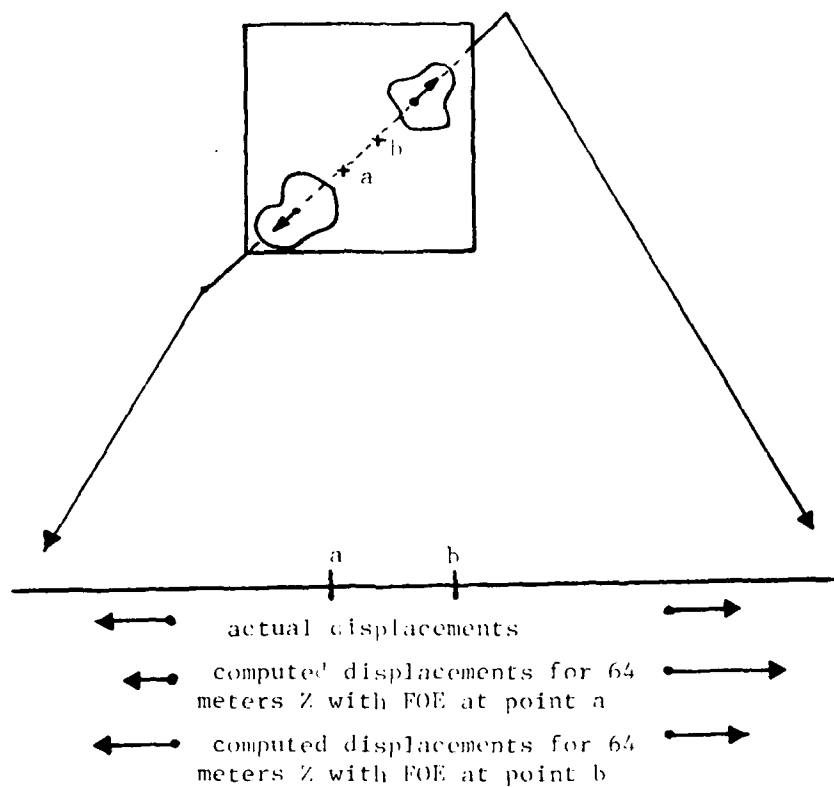
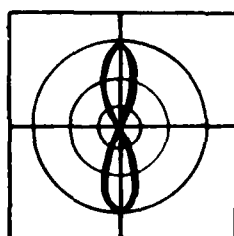
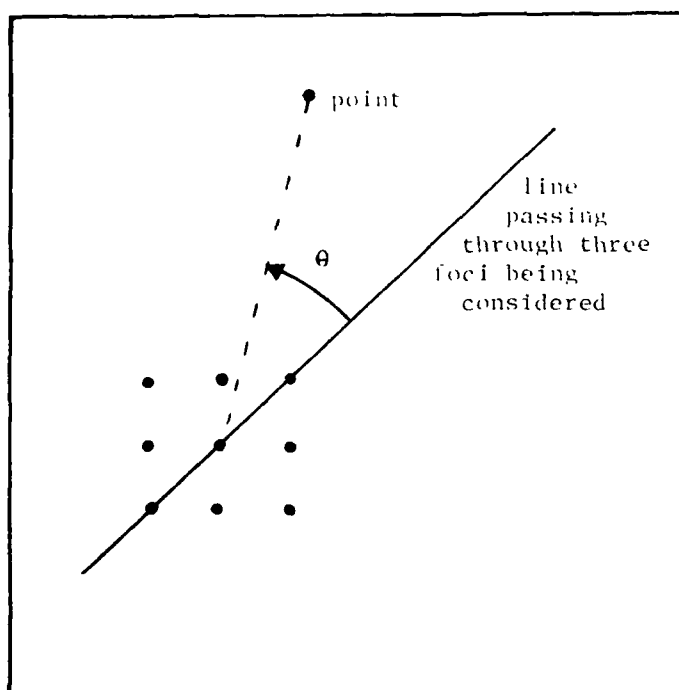


Figure 49. Simplifying by projecting along one line, it is obvious that asymmetries of surface distance can force the simple FOF search to a wrong result when the model is in error.



displacement of points lying on or near a line perpendicular to the line containing the three foci will show the greatest variation in terms of their direction, and therefore are more likely to disambiguate among alternative foci by contributing strongly to some error images when the direction of movement is incorrect.

Therefore, the error image produced for each of the three foci will be weighted so that points near the perpendicular line that passes through the central focus have more influence, while points near the line containing the three foci will have less influence. A sine function serves to weight the error because it is maximum at  $\pi/2$  (and  $3\pi/2$ ) and minimum at 0 (and  $\pi$ ) radians. After experimentation, the function selected was:  $E' = E \cdot (\sin^2 \theta)$ . The squared function is chosen to narrow the search locus, so that (radially) adjacent patterns would have minimum overlap.  $\theta$  is the angle between a line containing the point (passing through the central focus) and the line containing the three foci (see figure 50.)  $E$  is the point error computed by differencing the feature vector values, and  $E'$  is the resulting weighted error value.



polar plot of  
 $r = \sin^2 \theta$

Figure 50. The weighting function for the FOE weighted search. The weight for each error value is increased as that point nears the line perpendicular to the line containing the three foci under consideration.

This approach breaks up the nine foci into four groups of three each, where the central focus is in all groups. The error values obtained for the three foci of the different groups will be biased more strongly by different portions of the image. Since the error produced by each focus will be compared with all eight others to choose a minimum, the error value for each focus must be normalized.

To normalize the weighted error, the average error values obtained by using each of the three foci along a line are divided by the average error obtained for the central focus. Thus, the central focus will always have an error of 1.0, and the best choice during the weighted-error search is the focus producing the lowest normalized error. If the central focus is the best choice, all other foci will produce a normalized error greater than 1.0. Otherwise, the lowest will be less than 1.0.

III.2.6 Resegmentation. Sometimes surface discontinuities occur without giving rise to discernable visual characteristics in the image, and often visual discontinuities are imaged from sources other than those that occur because of surface discontinuities (see

chapter II, section 2). Static segmentations do, however, reflect many of the surface discontinuities which are discernable in images. But, it is impossible to guarantee correct surface segmentations directly by the analysis of a static image, i.e., there is not usually a one-to-one relationship between regions and actual surfaces.

When two or more surfaces at different distances are combined by the original segmentation into one region, they are incorrectly considered to be one surface. In this circumstance it is impossible for the Z and Y refinement process to determine a single correct value, since the region is the image projection of more than one surface. If a value of Z is correct for the nearer surface, it will be incorrect for the farther one, and vice versa.

When the assumed distance for a surface is incorrect, there will be an incorrect set of hypothesized displacement values. A comparison between points with different feature values results in error measures (in the error image) that are high. The size of a region of large error values is dependent on both the magnitude of the displacement error, and the size of the region. The

amplitude of the error depends on the magnitude of feature difference between the region and its surround (see figure 51). Note that an error will not be obvious if the two surfaces at different distances are visually similar.

The Z and Y search can be affected in two ways by an error in the initial segmentation. One is to converge to a value between the actual Z and Y values of the surfaces comprising the region. The other behavior is an oscillation between two values within the interval containing the minimum and maximum Z or Y value. Since the search is always terminated, we end up with a result that is somewhere near correct for at least one of the surfaces in question, or between the correct values for the surfaces.

The error image produced by the differencing of image points between the real image and the synthetic image based on the final Z and Y values, is used for resegmentation. If there are visible feature differences which indicate what areas are incorrect, then those areas will produce patches of large error values.

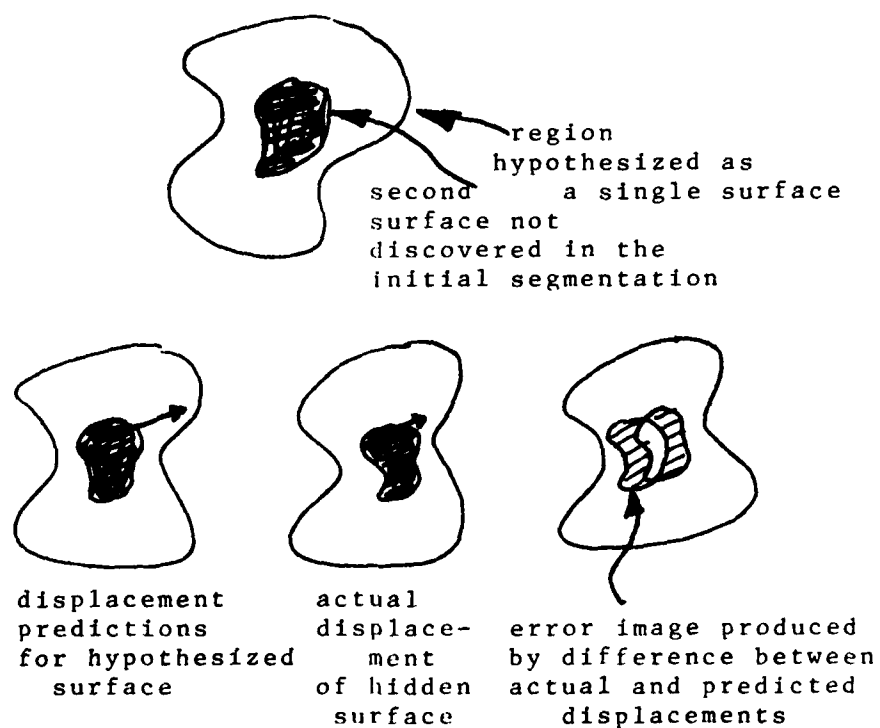


Figure 51. Resegmentation is based on contiguous areas of large error.

An automatic threshold selection algorithm (Kohler 1980) is used to produce new regions from the error image where the error values are above threshold. This threshold is chosen by measuring the contrast produced across all error value contour lines in the error image. The choice of thresholding algorithm, however, is probably not very important. Once the strongest error patches are roughly segmented from the error image, a plurality update rule (see appendix), similar to the one used for initial segmentation, is applied for two iterations to smooth the patches and remove those composed of one or two points.

The original segmentation is modified by adding the error regions in a replacement fashion. Wherever an error region exists all points in the original segmentation are replaced with a unique new label. All regions in the new segmentation are assigned Z and Y values equal to the average Z and Y of the model over their area. Thus, old surfaces retain their refined values, and the newly hypothesized surfaces begin with the supposedly incorrect values that led to their discovery. The search can now begin anew with the ability to more accurately model the motion of points on the newly hypothesized surfaces.

III.2.7 Surface merging. After the scene model is initiated and refined, the interpretation of the scene is available as regions that correspond to surfaces of one of two orientations. The values of Z or Y of each surface is specified, and adjacent regions with similar Z or Y values can be merged into one surface with the average Z or Y attached to the collection. We call this clean-up process a surface merge, for it merges excessively segmented surfaces.

We might also consider merging non-adjacent surfaces with the same distance. This, however, is not a good practice unless other information indicates that the two surfaces are indeed portions of the same surface. Suggested possible sources of information could be occlusion, spectral attribute similarity, and identical object identification. No experiments were done with non-adjacent merging.

III.2.8 Summary. Our system has been designed to interpret images in terms of a two-orientation surface model. It uses an hough transform test paradigm to refine a model which is initialized from static pictorial cues. The axis of travel is determined by an FOE search which can operate in two modes. Weighted-error is used for the



start-up problem, while a simpler search is suggested for use in attacking the continuation problem. With the FOE specified, image dynamics are used to refine the surface interpretation.

An hypothesize-test technique is used where image dynamics are predicted from a model of surface distance. This technique uses regions as the locality to be refined, and compares sets of pixel (point) features rather than the more commonly used edge features. The comparison between images uses interpolation to resolve displacements to sub-pixel accuracy. Errors in the segmentation used for the initial model are then discovered and the new model is refined again.

## C H A P T E R I V

### DATA AND RESULTS OF SURFACE INTERPRETATION

The surface interpretation system was developed through experimentation performed on a real moving image. These experiments demonstrate the effectiveness of some of the mechanisms in building an internal representation that depicts the physical environment. In each experiment a separate search algorithm is examined independently. Then the algorithms are joined together to form an integrated system.

This chapter is divided into three sections. The first section describes the data and the method of recording them. In it there is also a discussion of measurements taken at the scene and the deviation of the real camera from an ideal one. The second section presents the initial segmentations used for the experiments. The third section reports on a set of experiments that are designed to test the search and resegmentation processes. Recall that the search mechanisms derive the distances and heights of surfaces, and the correct location of the FOE. The process which resegments the image extracts surfaces that were incorrect in the initial model. Finally, the entire

system is tested, demonstrating its capabilities in handling the start-up problem.

#### IV.1 Data

Data were collected by taking movies from the passenger seat of a moving automobile. Over thirty minutes of movies were taken so that a short piece could be selected during which the automobile was moving through an interesting scene. The section selected was only three seconds long. This selective process was necessitated by the limitations of digitization, storage, and computational resources then available in our research laboratory.

IV.1.1 Collection and registration. Our scene was recorded in color with a super-8 movie camera which was mounted on a gyroscope and hand held by an automobile passenger. The camera was being displaced in the Y direction as the automobile traveled over bumps on the road, and in the X direction as the automobile was steered. A non-zero average tilt, pan, and rotation are present in the images as shown in figures 52 and 53.

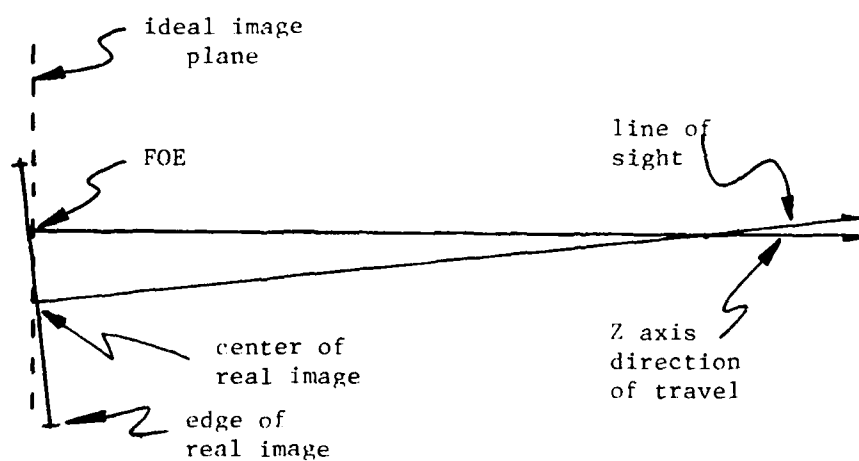


Figure 52. Actual pan and tilt of the camera was 2 degrees on average. This figure exaggerates the angle to 7 degrees.

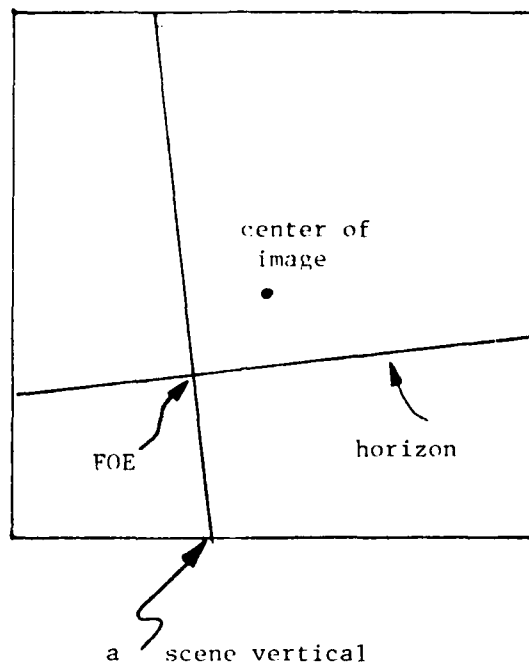


Figure 53. The camera was pointed slightly away from the direction of travel and was rotated by 0.75 degrees. This figure exaggerates the angles.

The moving image was taken at the end of September on a partly cloudy day at noon. Therefore, the trees are dark (since Autumnal leaves did not appear until October) and the scene is well lit. Because of the clouds, there was considerable visual texture in the sky.

A sequence of 54 frames was selected for analysis and digitized through red, green, and blue filters at 504 x 480 resolution with six bits of dynamic range per color (Pilipchuck 1979). The three color data images for each frame are in perfect registration. Only certain frames were chosen for analysis. Some experimentation was performed on every ninth frame; the experiments reported on here used frame numbers 45, 51 and 54, where 45 and 54 were one pair and 51 and 54 were another. The camera was operating at 18 fps, so two images that are nine frames apart would represent a 1/2 second interval.

The chosen digitized frames were registered by hand so that scene points far away from the camera would not appear to move when any two frames were successively displayed. The scene points used for registration were in the clouds and a distant road sign over 300 meters away and near the center of the image. The first part of the registration process involves translation where one

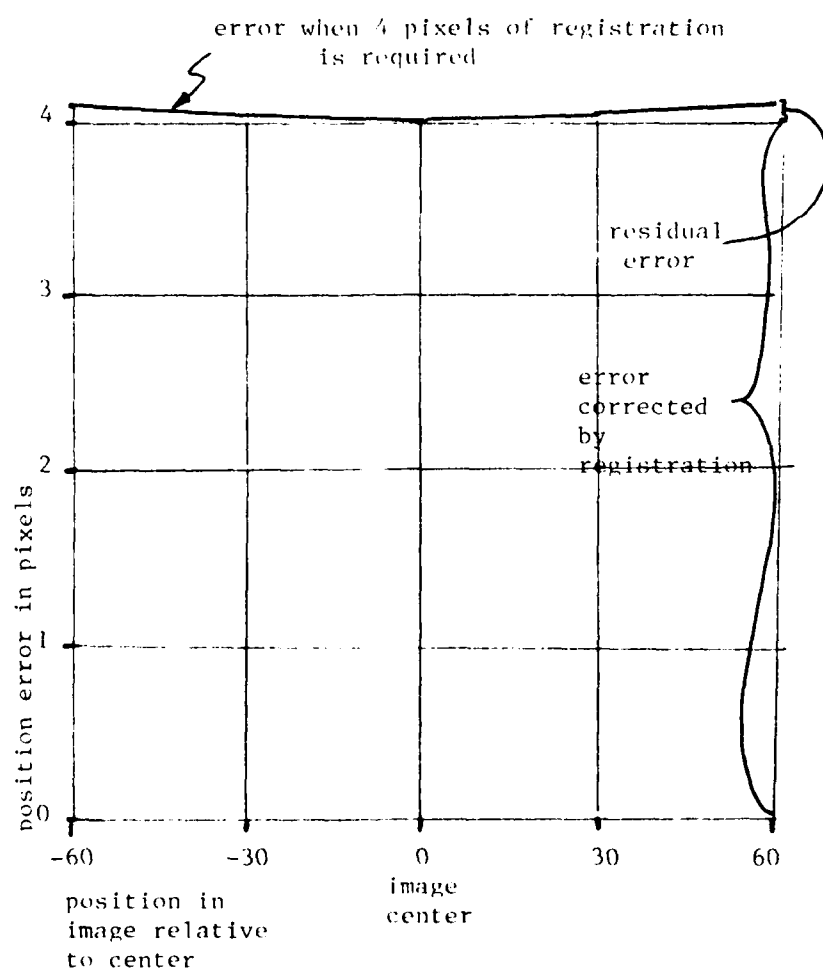


Figure 54. The position error in terms of pixels, showing the purely translational and residual errors.

image is brought into register with another by simple translation in the X and Y directions. The second part of registration is rotation, where one image was rotated (in the image plane) to register with the other. Only one frame of those selected exhibited appreciable rotation, and therefore required rotational registration. Correction for pan and tilt (beyond the act of registering the images) was considered, but not performed. Table 1 shows the registration required for the frames. The corresponding worst case residual error after registration (see chapter III, section 3.1.4) is shown in figure 54 and 55. The error was typically less than one tenth of a pixel in the worst case, and less in most cases. This low residual error, plus the inherent additional error that could be introduced by extra interpolation steps, led to the decision not to correct for pan- and tilt-induced projection distortion except for correction by registration.

Since it is desirable for the process to work not only on the whole image but also on any portion of it, two image sequences were prepared (see figure 56). The first is a 128 x 128 portion of the original image containing a road sign, a telephone pole, and a background tree. This subimage was selected because it



## REGISTRATION REQUIRED IN PIXELS RELATIVE TO FRAME #54

FRAME #	X		Y	
-----				
18	10	(2.5)	4	(1.0)
27	0	(0.0)	4	(1.0)
36	7	(1.7)	-7	(-1.7)
45	-4	(-1.0)	-14	(-3.5)
51	-5	(-1.25)	1	(.25)
54	0	(0.0)	0	(0.0)

Table 1. The registration required to align all frames with frame #54 is given here in pixels at 512 x 512 resolution, and at equivalent 128 x 128 resolution in parentheses. Registration was performed in discrete pixel units at full resolution.

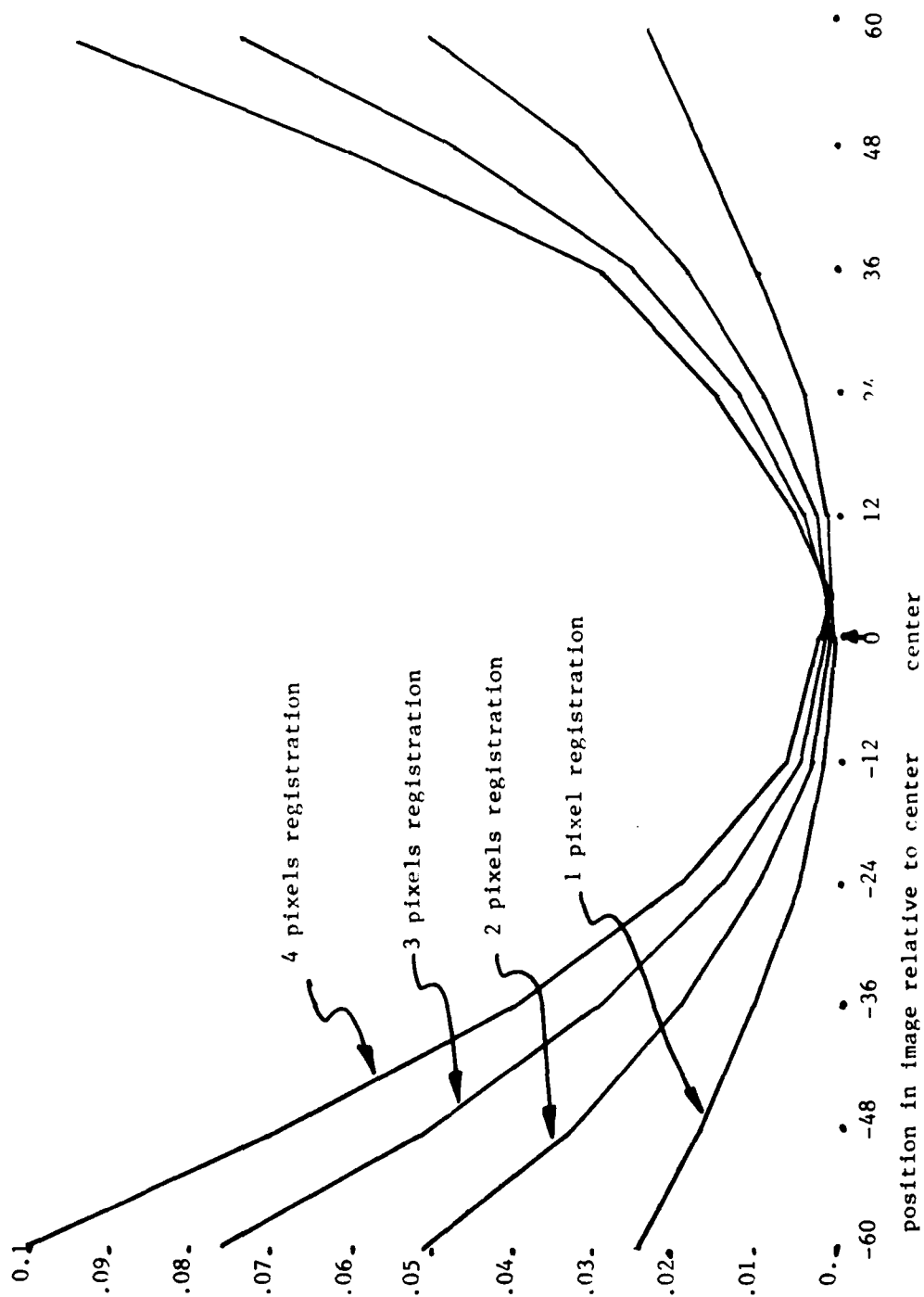


Figure 55. Residual error after registration. Note that full scale is 0.1 pixels.

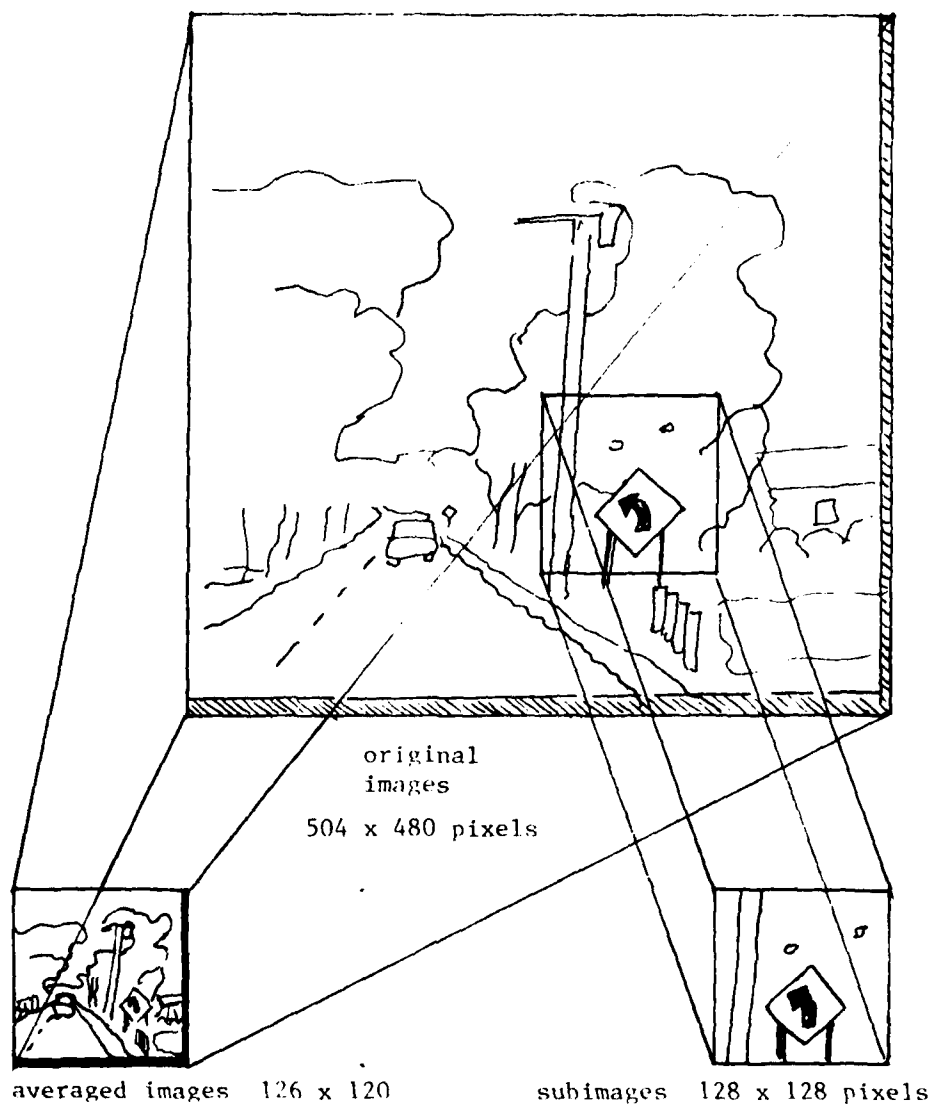


Figure 56. The data for the experiments were obtained by averaging the entire image, and by extracting a subimage.

allowed easy examination of the interpretation process where a small number of clear events are taking place. The second sequence is the entire image averaged to 126 x 120 using non-overlapping 4 x 4 windows. The entire image shows the sky, road, trees, the nearby sign, a distant sign, telephone poles, and some guard rail posts (see figure 57).

To develop and test the surface interpretation process two pairs of frames were chosen from the end of the 54 frame sequence. The first pair consists of frame numbers 45 and 54. This pair was used for both the entire image and the subimage. The second pair consists of frame numbers 51 and 54 for the subimage. The choice of 51 as the starting frame for the subimage was based on the fact that frame 51 had a naturally occurring segmentation "error" which posed some difficulty, and could be used to test the resegmentation process. Thus, frame 51 is actually a more difficult frame to deal with (see section IV.2.1). Tests were run on images from the last part of the 54 frame sequence because there were some large objects with considerable image velocities. In addition, the system was run on every ninth frame, representing half-second intervals, from numbers 18 to 54. The results for the entire sequence were not

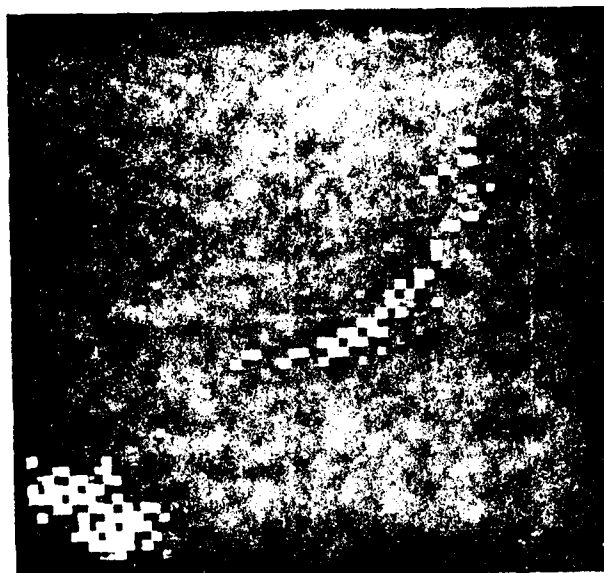


Figure 57. The whole image frame 45  
averaged

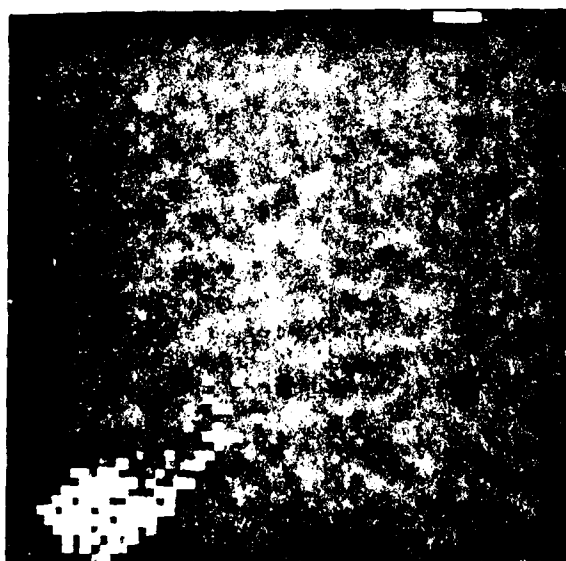
analysed in detail, but rather presented to indicate that the process delivers similar results on other portions of the sequence.

IV.1.2 Scene measurements. To evaluate the experimentally derived distances and heights, actual scene measurements were obtained. The computation of the distance to scene points (given the corresponding displacement on the image) relies on the parameter  $\Delta Z$  to scale the results (equation 5 chapter III). Therefore, we had to measure both the parameter  $\Delta Z$ , and the disposition of actual surfaces with respect to the camera.

To accomplish this measurement, prints of the movie frames and the camera used to take the movie were taken to the scene. The photographic prints were visually matched with the viewfinder image. Multiple instances of occlusion of objects gave clear reference points that could be used to place the camera. This resulted in a placement of the camera to less than  $\pm 1.0$  meter for any individual frame.



a)



b)

Figure 58. Two-dimensional feature space histograms of a) whole image and b) subimage. Horizontal axis is the V color feature and vertical is the W feature.

The centers of the possible locations for two of the frames chosen for experimental analysis (numbers 45 and 54) were eight meters apart. This yields a  $\Delta Z$  of .88 meters per frame. This value was verified by measurements of 0.86 to 0.90 meters per frame across other sequences of the 54 frames using the same technique. The average from frame one to 54 was also .88 meters per frame. Since the automobile velocity and frame rate were very nearly constant, we assume that the figure of .88 meters is correct to  $\pm$  five percent. This figure is derived by assuming  $\pm$  1.0 meters of uncertainty at both the beginning and end of the 54 frame sequence.

The actual distances and heights of scene surfaces could then be measured directly from the position that the last frame was imaged. These measurements are presented with the experiments (see section IV.3) where the derived Z values can be evaluated. The derived Z values must be judged within the context of the  $\pm$  1.0 meter error in camera placement, and a possible error in measuring the parameter  $\Delta Z$  which is probably less than  $\pm$  five percent. These two uncertainties are manifest in an absolute uncertainty of distance to  $\pm$  1.0 meters plus a multiplicative factor of  $\pm$  0.05.



The value for Y is measured from the Z axis. Under the assumption that the axis is parallel to the road, the Y value for the road is -1.1 meters; however the road is slightly cambered, to about -1.2 meters at its edges. Some ground plane is visible to the right of the road. This grassy area is slightly lower than the road in the foreground, about -1.5 to -2.0 meters of Y. In the background it begins to rise again, but the exact amount is hard to estimate.

#### IV.2 Segmentation

The first step in forming a surface model requires the segmentation of the frame at time  $t_0$  (for our first analysis this will be frame 45). This segmentation is used as the initial set of hypothesized surface projections, and therefore defines the localities over which comparisons between real and synthetic images are made.

The initial features were selected (interactively) to provide good separation of distinct visual properties of the image. The V and W color features (see appendix) were found to serve this purpose. Either of these features would be sufficient to define some histogram clusters, and in turn, some interesting regions in the

subimage. However, the complexity of the full image demanded greater discrimination than a single feature would allow. For consistency, rather than the use of one feature on the sub-image and two features on the averaged image, the two dimensional histogram  $V \times W$  was chosen for both the subimage and the entire averaged image.

The initial segmentations were formed in four steps, as detailed in chapter III, section 3.1.6. In the first step the point features,  $V$  and  $W$ , were computed and a two-dimensional histogram was formed. Four clusters, as described below, were found automatically in the subimage, and five were found automatically in the averaged full image. The feature image was then formed, where each pixel is a vector of normalized cluster distances. In the second step the maximum label from the feature vector of each pixel was assigned to that pixel, reducing the vectors of possible labels to a single label at each pixel. This is effectively a minimum distance classifier applied to the feature value of each pixel, where the cluster centers in feature space are the target classes. The third step was the application of two iterations of the plurality update rule, where the majority label in each pixel's neighborhood becomes the new label at the central pixel (see appendix). The

AD-A107 565

MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INF--ETC F/G 9/2  
COMPUTER INTERPRETATION OF A DYNAMIC IMAGE FROM A MOVING VEHICLE--ETC(U)  
MAY 81 T D WILLIAMS  
COINS-TR-81-22

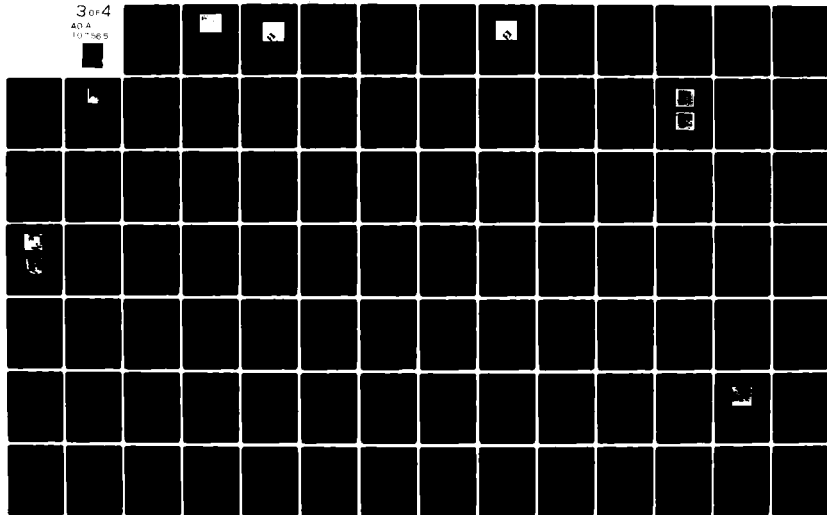
N00014-75-C-0459

NL

UNCLASSIFIED

3 of 4

40 A  
10-75 5



fourth step was region labelling, wherein a unique region number is assigned to each region of contiguous labels.

One of the experiments presented below is based on the segmentation derived automatically from these features. However there are several interacting problems, any one of which could cause system failure. Since our approach is to test the subsystems in isolation, some of the experiments should have an accurate initial segmentation. Therefore we have chosen to provide an initial segmentation for the subimage which was derived through two steps. First, a segmentation was derived automatically, by use of a plurality update rule. Then, the regions were combined manually so that the final segmentation was a reasonably good approximation to the scene surfaces. The experiment that was run on the entire averaged image shows the performance on a segmentation generated directly by the computer where there was no further modification by hand.

IV.2.1 The subimage. The subimage was used as data in the first four experiments. These experiments demonstrate that the surface distance search subsystem - the Z Search - could achieve the sub-goal for which it was designed. The particular subimage was chosen because



Figure 59. The four feature images which comprise the four element feature vector image used for segmentation and inter-image comparisons. Frame number is 45.



Figure 60. Subimage for frame number 45

of the clarity of the events that occur in it.

Four clusters were found in the histogram of the first subimage. (see figure 58). These correspond to 1) the yellow portion of the sign, 2) the telephone pole and some of the border around the sign, 3) the lighter portions of the tree texture and some of the border of the sign, and the border between the arrow of the sign and the yellow background, and 4) the darker portions of the tree and the central part of the arrow of the sign. (see figure 59).

The experiments which used the subimage required a good initial surface segmentation so that experiments could be conducted on subsystems without extra complications. Thus, the 77 regions obtained by this automatic process were coalesced interactively into three regions. This resulted in a rough separation of the sign, the telephone pole, and the background tree (see figure 60), making this initial segmentation (figure 61) close to correct. To test the resegmentation subsystem, frame 51 was selected as the  $t_0$  image because it generated a segmentation which joined the sign with a piece of a building which was visible through the background tree. The segmentation for frame 51 was





manually modified, but the segmentation error was left unchanged (see figures 62 and 63).

IV.2.2 The averaged image. Experiment #5, described below, was designed to test the entire surface interpretation system on the full image. For this treatment the full image was averaged by non-overlapping 4 x 4 windows, resulting in a 126 x 120 image. Considerable loss of resolution results from this averaging, especially noticable around the edges of areas with visual contrast, and in areas with strong texture. In these areas, the transitions from dark to light have been blurred. Images with reduced sharpness are very difficult to handle with systems that rely on inter-pixel differencing and discrete edge feature placement, and therefore the averaged image offers a good test for this system, where the use of motion information can provide recovery from ambiguous or erroneous segmentation decisions.

As with the subimage, the V x W histogram was used to find clusters. Five cluster centers were automatically selected, and the minimum distance classifier found 232 regions. Two iterations of the plurality update rule (see appendix) reduced this number



Figure 62. Subimage for frame number 51

STATION: 000000 000000 000000 000000 17 49 30 2-APR-61  
C 127 0 000  
1 000000

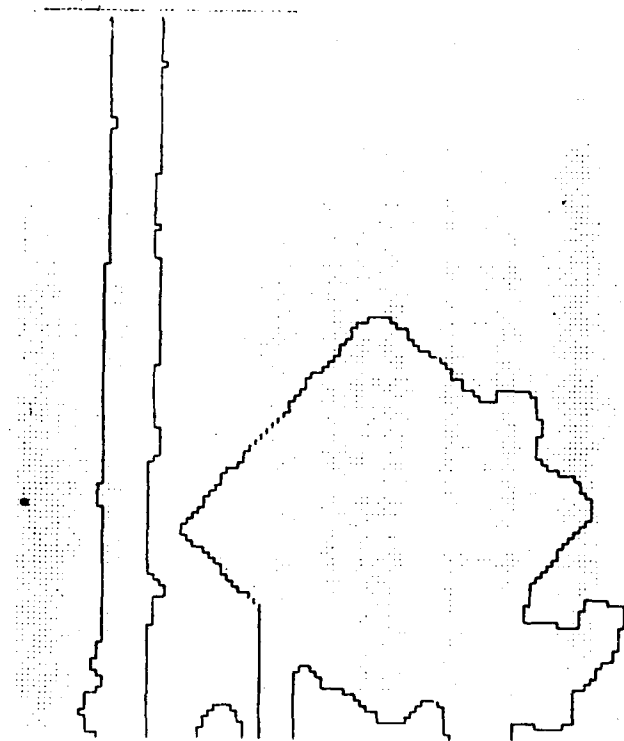


Figure 63. The segmentation used for subimage frame number 51. Note the inclusion of background bright spots which are not part of the sign.

to 152 regions. These regions were not modified further (as were those in the previous experiments). Instead, the segmentation produced by the computer was used directly, without human intervention (see figure 64).

#### IV.3 Experiments

Five experiments are presented that summarize the performance of the various subsystems of the surface interpretation system. The three mechanisms that were examined are the Z and Y search, the FOE search, and the resegmentation process. The first three experiments examine each of these three subsystems under the important assumption that proper data are provided by the other subsystems. In particular, we consider Z search with a correct FOE, FOE search with correct values of Z, and resegmentation of a mostly correct segmentation and surface model. This will permit fair evaluation of each mechanism without requiring their coordination. The fourth experiment was designed to explore the interaction of the mechanisms for FOE and Z search. The fifth experiment tests the functioning of the entire system with minimum human intervention.

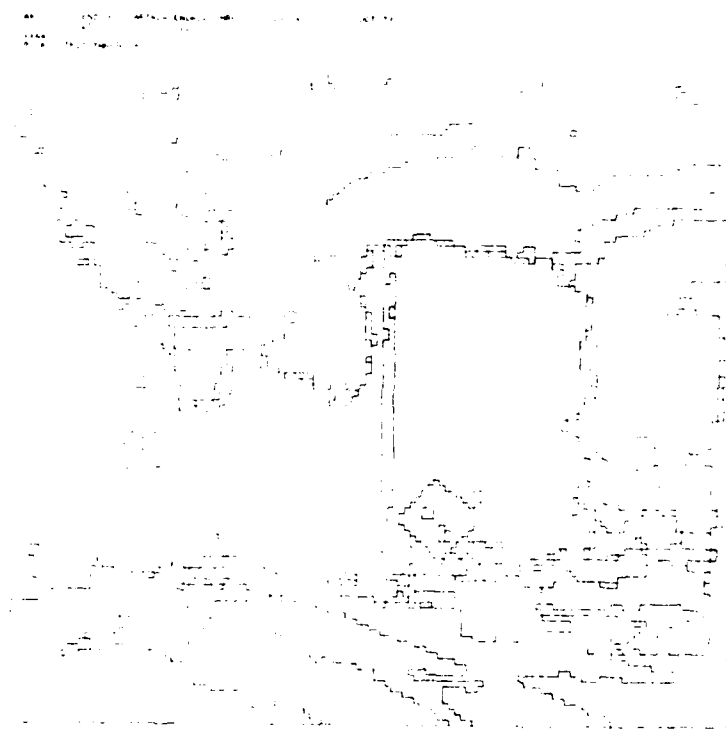


Figure 64. Segmentation used for image pair # 4

IV.3.1 Experiment # 1, search for Z, given: correct FOE and generally correct segmentation. The purpose of the first experiment was to evaluate the search mechanism for determining the distance of a surface, under the assumption that the correct FOE is known. This process for determining surface distance is referred to as "Z search". All surfaces were initially assigned a single (in most cases incorrect) initial Z value. In this experiment, various such initial models were tried in an examination of the search behavior, but we report on only one experiment that typifies the search events. The results from applying Z search to these various initial models were almost identical.

The FOE was placed at a point that lay on the passenger side of an automobile in front of the one carrying the camera. This was a logical choice for the FOE since the car in front of ours should have been in a position that ours was about to occupy. This FOE was confirmed by drawing several lines on the superimposed images to track distinct scene points. These lines all passed within six pixels (on a 512 by 512 grid) of the chosen point.

There were no horizontally oriented objects in the subimage. Therefore, we limited the processing to the discovery of a Z value for each surface assuming vertical surface orientation.

GOAL: To determine the value of Z associated with visible portions of surfaces in the subimage, where all surfaces are assumed to be in the vertical orientation.

GIVEN: A correct FOE, and an initial segmentation that is mostly correct. The Z value for each surface was initially set to 512 meters.

RESULTS: After twenty iterations of refinement all k values had attained the stopping criterion of .025, i.e., a search increment of  $\pm 2.5\%$  for each surface Z value. Figure 65 shows the progression of the search. The resulting Z values for the surfaces (see figure 66) were 34 meters for the sign, 47.4 for the telephone pole, and 61.8 for the tree. These values are quite close to the correct values. The actual values of Z when the last frame was imaged are 34 meters for the sign, 45 meters for the telephone pole, and 55 to 65 meters for the tree (various visible parts of the crown are at different distances).

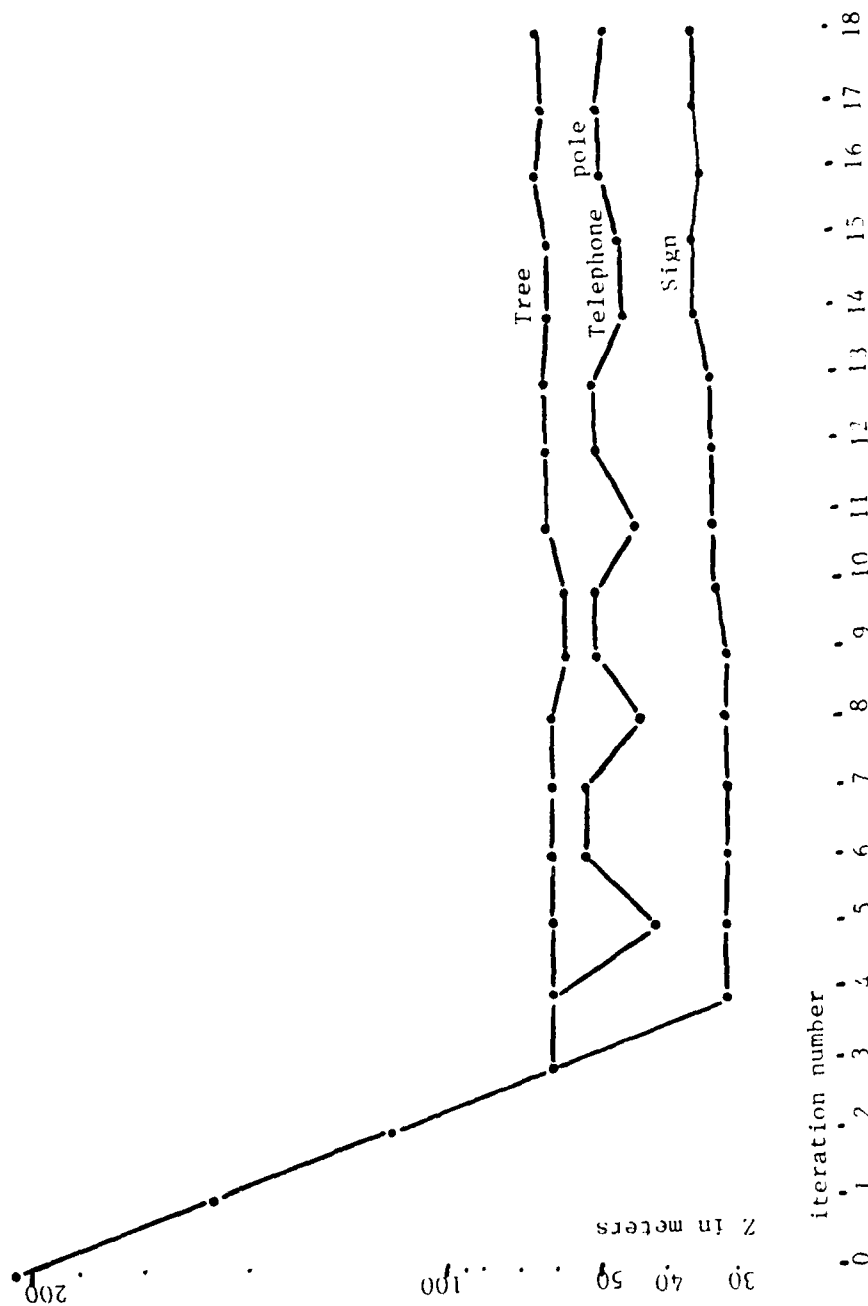


Figure 65. Experiment #1 led to a model for the three surfaces in the subimage.





Figure 66. Surface model for the subimage.  
Brightness encodes distance.

Recall that an error value is derived for each surface value for each surface. The error value is the average (across the surface) of the difference between the actual and synthetic  $t_0$  feature images. This value runs between 0 and 64, where 0 is no error, and 64 is a maximum. This range results from the scaling that was performed on the initial features to generate a two-dimensional histogram. In practice no minimum went below 2.0 and no maximum above 40. To understand the performance of the search we have plotted error value versus Z value. Figure 67 shows the error functions for the surfaces in these images. The figure shows details of the error function obtained by trying a number of hand selected Z values. Note that for these surfaces the error functions have clear minima, and the error values increase sharply for the nearly flat surfaces, while it rises very slowly for the tree which is really composed of a distribution of distances.

Figure 68 indicates the reduction of the error values during progression of the search. This figure depicts the error values obtained for the Z values that the search generated during successive iterations.

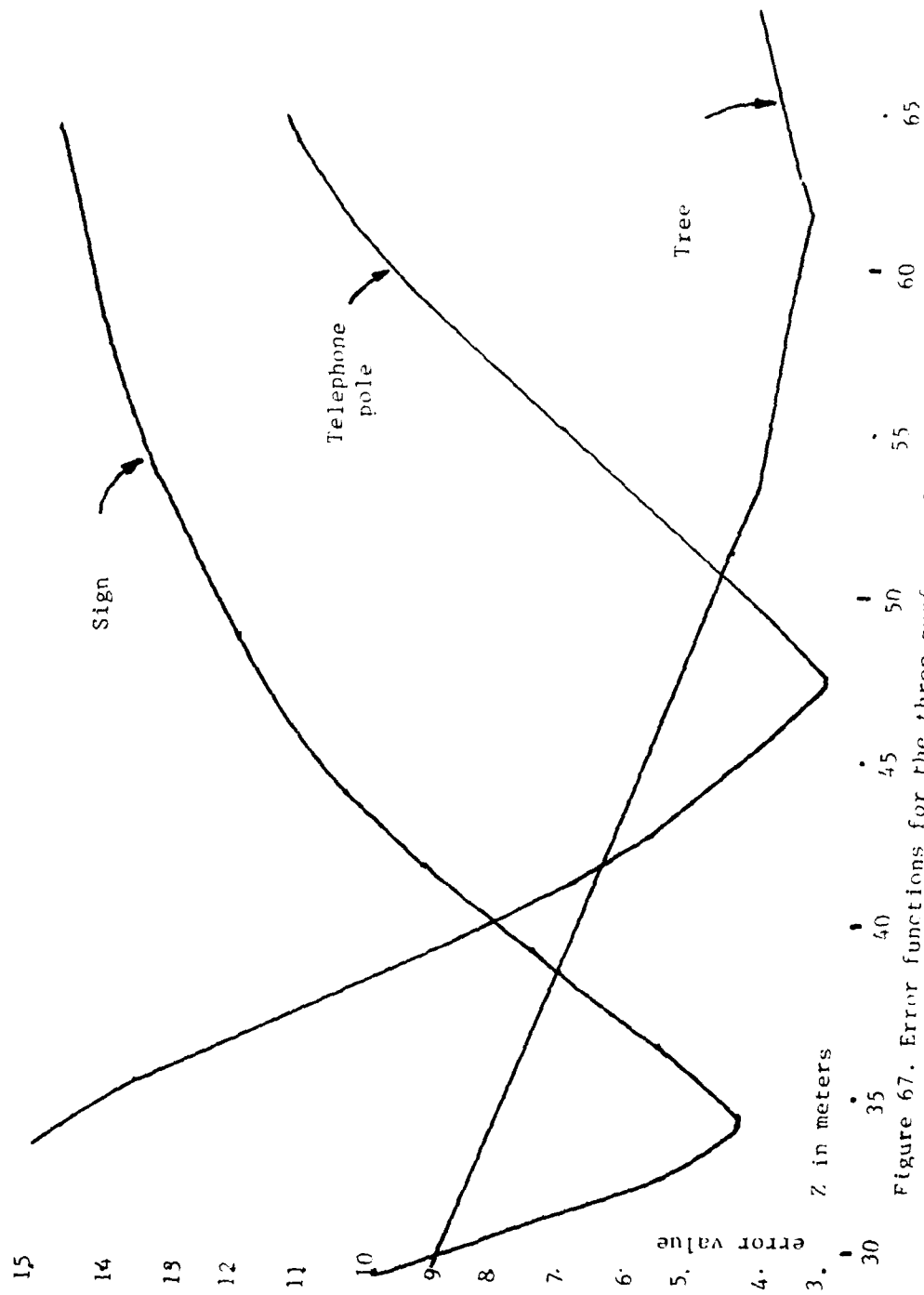


Figure 67. Error functions for the three surfaces of the subimage

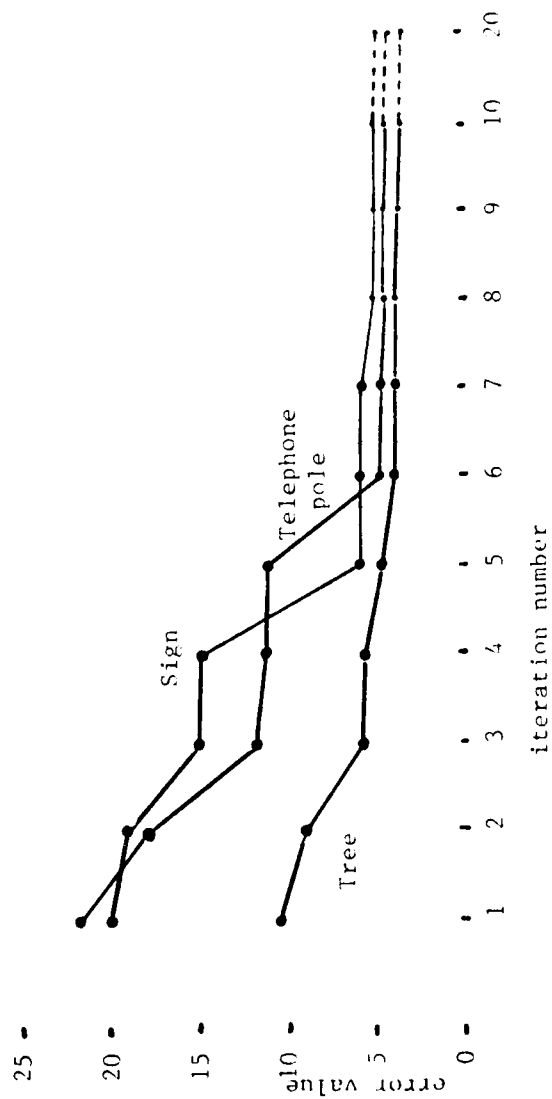


Figure 68. Error was reduced as the search progressed. Here the error values for the three surfaces are shown for the first ten and last iteration.

DISCUSSION: These promising results indicated that the distance to surfaces, i.e., the Z values could be obtained automatically with the search process, at least when there are available both a reasonable segmentation which approximates surfaces of the scene, and an accurate FOE. The resulting distance values are within experimental error bounds, and the error functions are reasonably well behaved.

IV.3.2 Experiment # 2, unweighted search for FOE, given: a refined surface model. This experiment, again using the subimage, was intended to evaluate the FOE search process in a controlled manner. Here, the Z values of each surface were fixed at the result from experiment #1. Thus, the FOE search proceeded with the assumption that a good model was available. Because of this assumption, the simple (rather than weighted) FOE search mechanism could be tested.

GOAL: To demonstrate the effectiveness of the unweighted FOE search applied to the subimage and operating with a correct surface model.

GIVEN: A reasonably accurate surface model (which was actually developed with the use of the correct FOE, i.e., the result of experiment #1).

A starting point for the search was chosen near the center of the original image, at the (0,0) coordinate of the subimage (upper left hand corner). The search would take eight steps to achieve a  $\pm 1.0$  pixel value for  $k$ , the minimum refinement increment. A ninth iteration is also performed to check that the final focus has less error measure than the eight adjacent foci.

RESULTS: After nine iterations the FOE was found to be (344, 152). This differed from the one used in the first experiment by only one pixel (out of 512) in the Y direction, and no difference in the X direction (see figure 69). Table 2 summarizes the search, where coordinates are measured relative to the origin of the subimage.

DISCUSSION: This result indicates that the FOE search located the FOE when a correct surface model was employed. Unfortunately, it does not show that the FOE it finds is correct, only that it finds an FOE that fits a refined distance model.

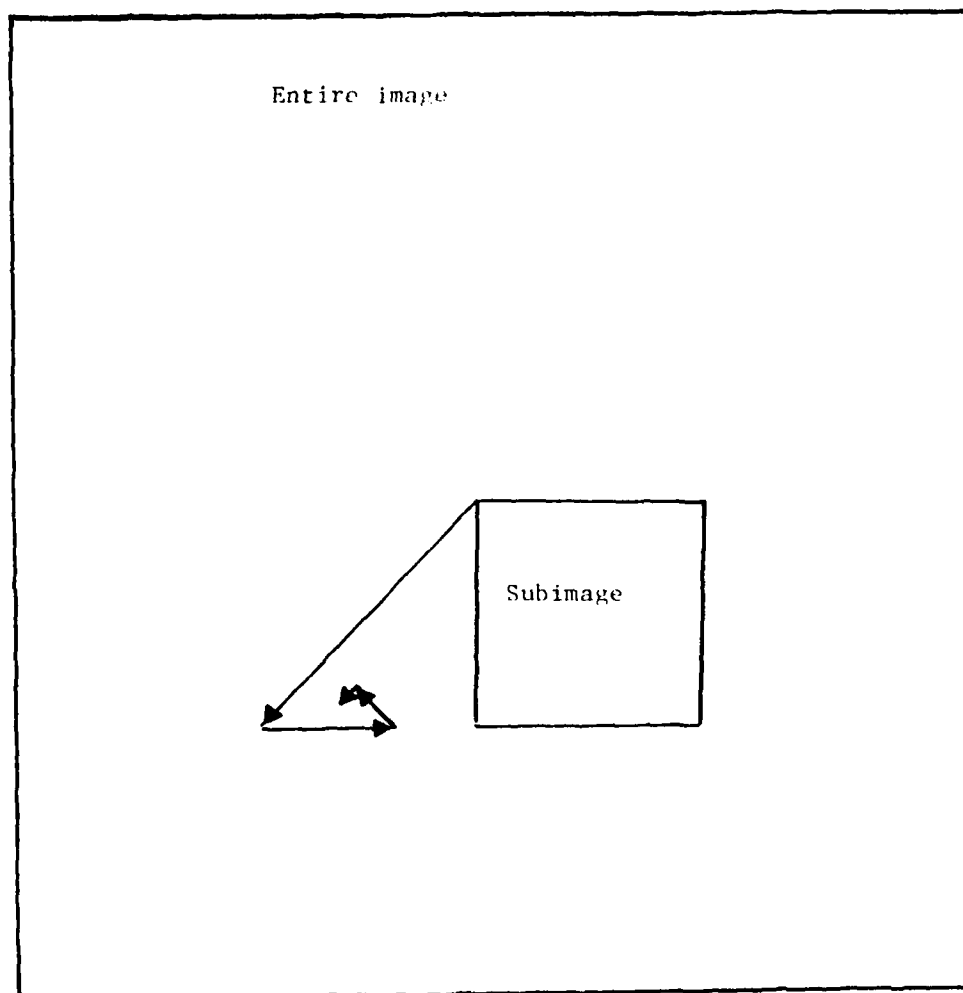


Figure 69. The FOF search was performed using only the data in the subimage pair. This figure represents the search path.

ITERATION NUMBER	Search for FOE			END		DIRECTION
	BEGIN ROW	COL	INCREMENT K	ROW	COL	
1	0	0	128	128	-128	↖
2	128	-128	64	128	-128	.
3	128	-128	32	128	-96	→
4	128	-96	16	128	-96	.
5	128	-96	8	120	-104	↙
6	120	-104	4	124	-104	↓
7	124	-104	2	124	-104	.
8	124	-104	1	124	-103	→
*9	124	-103	1	124	-103	.

Table 2. The successive iterations of FOE search. a dot under "direction" indicates that the central focus was chosen. The ninth iteration checks to ensure that the last focus is surrounded by foci with larger error measures.



IV.3.3 Experiment #3, resegmentation, given: a refined model. The third experiment tested the resegmentation process in an attempt to remove areas of the segmentation that are in error. The segmentation in the first experiments was manually corrected; consequently, it had very little error in order to provide a basis for evaluation and control over the processing. Thus, there were only a few errors in the initial model that needed to be attended to in this image. However, by choosing image numbers 51 and 54, a serious segmentation error does occur. By allowing this error to remain when interactively modifying the segmentation, we can demonstrate the capability of the resegmentation process.

Recall that an "error image" is an image where each pixel is given a value equal to the difference between feature values of that pixel in the real  $t_0$  image and in the synthetic  $t_0$  image. The synthetic image represents the inferred position based upon hypothesized surfaces in the model. The error image that is based on the refined model is used for resegmentation.

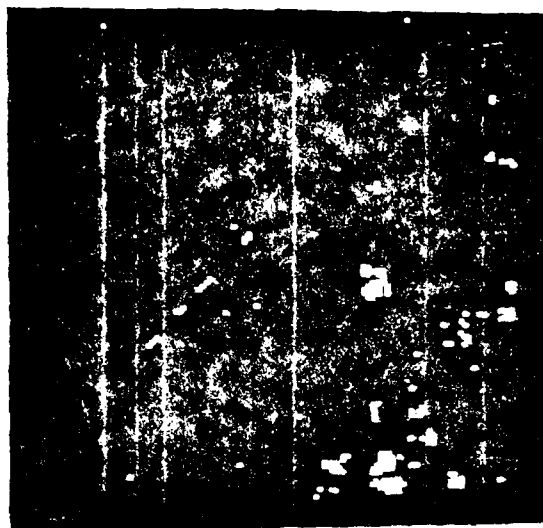
An automatic threshold selection algorithm (Kohler 1980) is used to produce new regions where values in the error image are above a threshold. The selection of an

appropriate threshold is based on the measurement of differences between adjacent pixels in the error image. The threshold which produces the maximum average difference is selected. Thus, the strongest error patches are roughly segmented. Then the plurality update rule (see appendix) is applied for two iterations to smooth these patches and remove those composed of only one or two points.

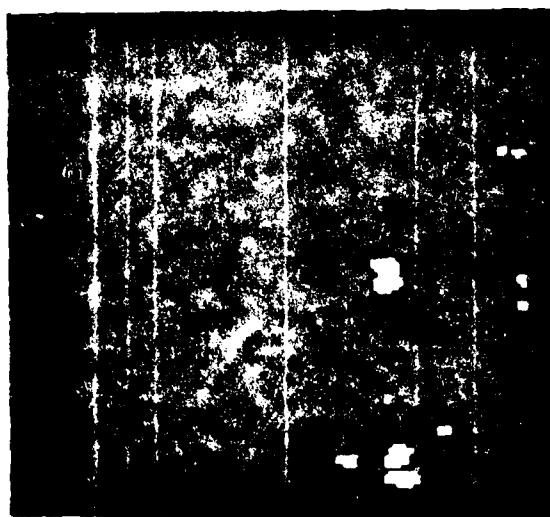
GOAL: To resegment the subimage based on an error image, and to derive correct model information for the areas previously assigned incorrect values of  $Z$  because of an incorrect initial segmentation.

GIVEN: A model with an incorrect segmentation, where some hypothesized surfaces are actually several surfaces at different distances.

RESULTS: As shown in figures 70 and 71, the thresholded areas correspond to surfaces that were incorrectly segmented. All of these areas were more distant than the initial model search indicated. Figure 71 shows the final model segmentation. The blob adjacent to the upper right of the sign as well as those on the lower right are holes in the tree showing the side of a distant building.



a)



b)

Figure 70. a) thresholded error image for images 51 and 54, and b) resulting error regions after two applications of the plurality update rule.

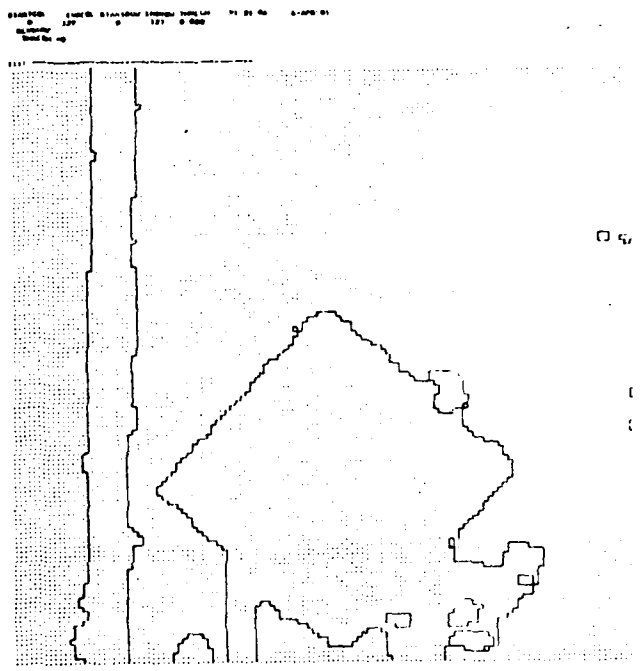


Figure 71. Model after re-segmentation and merging from frame numbers 51 and 54.

DISCUSSION: This experiment shows that when sufficient pictorial contrast (exhibited by feature difference) exists between areas which have been incorrectly joined as a surface (so that the hypothesized Z value is incorrect for at least part of the hypothesized surface) it is possible to segment such an area from the image, and proceed to search for its correct Z value.

Sufficient visual differences existed between many of the predicted and actual surfaces that were in error to completely extract them as regions. As explained in chapter II (in tracking techniques), this is not generally the case, and the portions of the incorrectly segmented lower right portion of the sign that did not have sufficient contrast could not be segmented. Often in the case of relatively uniform regions, only the leading and trailing areas of such surfaces will be detected. However, repeated application of the resegmentation process can reduce the error regions until they are completely removed and refined.

IV.3.4 Experiment #4, unweighted search for FOE, given:  
a uniform (mostly incorrect) Z model. In preparation for the marriage of FOE search and Z value search, we performed the fourth experiment. The first two experiments showed that the FOE search could proceed with a correct model of Z values, and a Z value search could proceed with a correct FOE. The aim of this experiment was the evaluation of the performance of the unweighted FOE search with the model grossly in error (most Z values wrong). We expect this technique to fail, for reasons explained in chapter III, thereby justifying the need for the weighted FOE search mechanism. Performance was judged by examining the first few FOE search steps with different initial uniform models of surface distance.

GOAL: To demonstrate the limitation of simple FOE search with an incorrect uniform model.

GIVEN: An initial model with correct segmentation, but incorrect uniform Z values, and a starting FOE at the center of the original image.

RESULTS: As summarized in figure 72, a uniform Z value of 256, 128, 64, and 32 meters all produced a first step of the search for the FOE that was correct, i.e. the closest step toward the solution given the initial

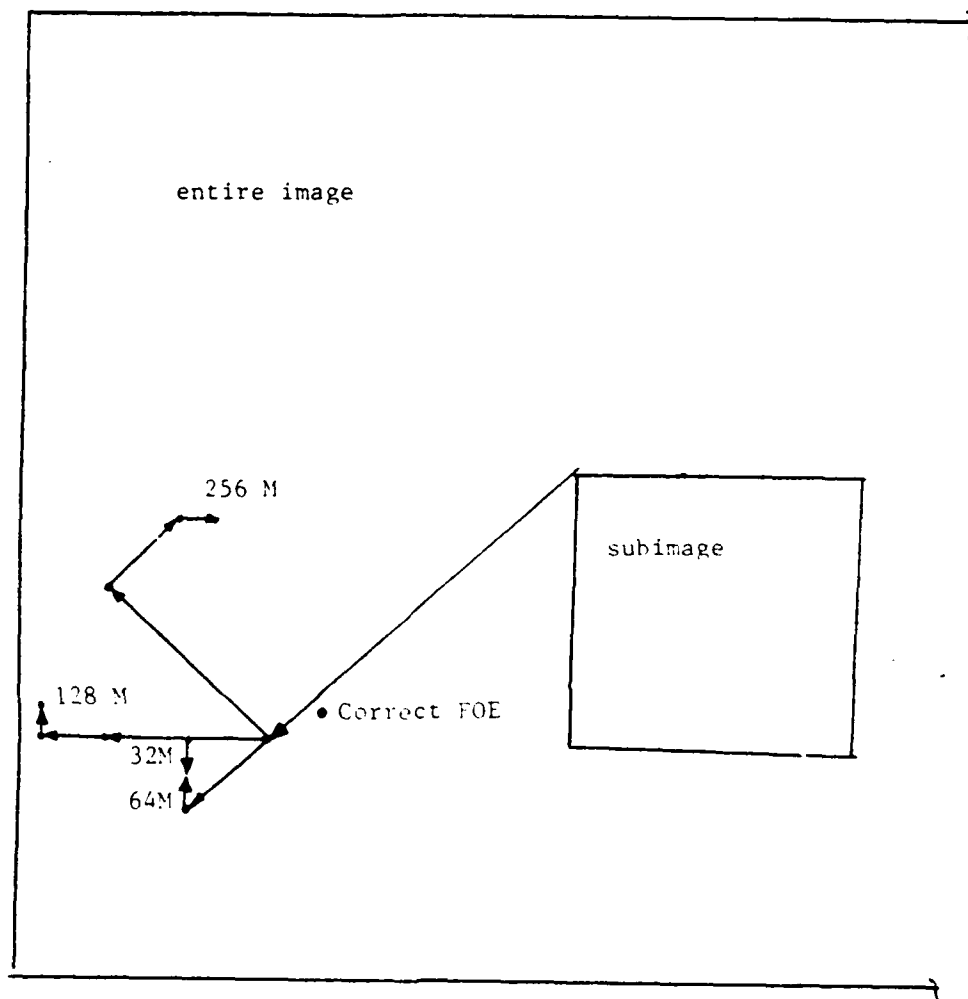


Figure 72. The unwieghted search using initial models with 256, 128, 64 and 32 meter settings.

step increments. However the second iteration of FOE search was grossly incorrect for the 256, and 128 meter settings. For these settings the search continued to diverge from the correct solution. The first five iterations for the 32 and 64 meter settings remained in the vicinity of the correct solution, rather than diverging from it, although they produced different paths to the same final value. Table 3 summarizes the searches. In this table the direction arrows represent the corresponding directions in (figure 72) of FOE movement during each iteration of the search..

DISCUSSION: Since 32 and 64 meters are reasonable estimates for many surfaces in the subimage, it is not surprising that such choices would produce the best FOE search. When the model is further in error, the FOE search moves further from the correct solution.

The results show however, that the search cannot continue beyond two steps without going wrong (converging to an incorrect result), even for the best initial model (64 meters). Since the goal is to achieve an interpretation with little or no a-priori knowledge, the choice of 64 vs. 128 or 32 meters should be inconsequential. This, result demonstrates the necessity

















UNIFORM Z SETTING IN METERS	DIRECTION RESULT FROM EACH ITERATION				
256					•
128					•
64		•			•
32		•			•
ITERATION #	1	2	3	4	5

Table 3. Successive iterations of unweighted FOE search run with various uniform settings of Z. A dot indicates that the central focus was chosen.

of the weighted FOE search.

IV.3.5 Experiment #5, weighted search for the FOE and surface model simultaneously, given: a uniform (mostly incorrect) Z model. The remaining problem, as demonstrated by experiment #4, is the major goal of this research. That is, the unification of FOE search and model refinement into a surface interpretation system that could produce a reasonable model with limited prior knowledge. We proceed by first obtaining an initial static segmentation; second, selecting a uniform set of Z and Y values; third, alternatively applying weighted FOE and model search steps; and finally, resegmenting based on remaining error. Since the entire image was used for this experiment (by averaging the original), the ground plane was visible, and the model included values for Y as well as values for Z. Since the entire image is averaged, the FOE search is on a 128 by 128 grid and only takes six steps to complete.

The strategy of alternatively applying weighted FOE and model searches is based on controlling the area of the image over which the model search could be driven to choose wrong Z or Y values. Since the goal is to keep the FOE error from driving the Z and Y search to produce

wrong values, and because the incorrect model can drive the FOE to an incorrect result, we must first apply as many FOE search steps as can be relied on that converge toward the correct solution before proceeding with Z and Y search. Then, we must understand the nature of the errors caused by incorrect placement of the FOE, and use this information to control the Z and Y search.

Recall that the relation between Z and  $\Delta d$  is expressed in equation (4). Now consider the effect of an error in placement of the FOE on the value of Z that the search will converge upon. If the FOE were incorrectly placed further from or nearer to an image pixel, it would lengthen or shorten (respectively) the values of d and therefore  $\Delta d$ . Thus, for a given error in FOE, and a given Z, pixels closer to the FOE would have a greater induced error in  $\Delta d$  than those further from the FOE. It is possible to express the error in terms of Z resulting from FOE placement error thus:

$$e = \frac{\text{FOEerror}}{d},$$

where e is the multiplicative model error term arising from FOE placement error divided by the distance d of a

point from the actual FOE.

To control the model search with the FOE in error, we choose a stopping criterion  $k$ , and do not allow the model search to proceed beyond it. Then, surfaces which have an FOE induced error less than the chosen  $k$ , will have their search stopped before an attempt is made that could result in a wrong choice.

We can control the FOE error under that assumption that the FOE search is proceeding toward the correct result. If so, the increment (in pixels) of the next FOE search is the average error that the present FOE has with respect to the final FOE. For convenience, the units of the fraction can be pixels, and the error a percentage.

If we wish to prevent incorrect FOE placement from introducing  $Z$  and  $Y$  value errors in all but  $1/16$  of the image (in the worst case), then the value of  $d$  would be 16, because a  $\pm 8$  by  $\pm 8$  area around the actual FOE contains  $1/16$  of the image points. For the first FOE search, three iterations (as far as the FOE search can be expected to give correct results) would make the next  $k$  equal to four pixels, yielding an average error of four pixels for FOE placement. Thus  $e = 1/4 = 0.25$ .

By running the model search with a stopping increment of 0.25 we are assured that, in the worst case, model errors might exist in less than 1/16 of the image. Then, another application of FOE search, resulting in uncertainty of result to two pixels, yields a stopping criterion of  $k=.125$ , for the next model search. Another application of FOE search (resulting in uncertainty of 1 pixel) is followed by model search with stopping criterion of  $k=.06$ . This is followed by a final FOE search step (resolving the last pixel of placement).

Experiment #5 was performed on the entire image to observe the effect of choosing various (generally incorrect) uniform Z values. Out of the set of 128, 64, and 32 meters we will note differences in final FOE placement. The starting Y value was 1.1 meters.

GOAL: To demonstrate that a sequence of search steps can result in the simultaneous derivation of a correct FOE and surface model.

GIVEN: An initial segmentation and uniform values of Z and Y for each hypothesized surface.

RESULTS: One application sequence was found to be very effective at both finding the FOE and refining the model (see figure 73). Figure 74 and table 4 summarize the FOE search, and figure 74 shows the number of surfaces undergoing change (there are 152 surfaces) in value during the model search iterations.

DISCUSSION: By using weighted FOE search the effects of model errors on FOE placement were greatly reduced. Three steps of the weighted FOE search generated the same results (of FOE placement) for initial models with uniform Z values of 128, 64, and 32 meters (see table 4). The accuracy of FOE placement allowed the model to be driven to a stopping "k" of .25 with possible errors in 1/16 of the image. This took eight Z and Y search steps. Then one more FOE search step was followed by 12 model search steps with the stopping criterion set to a "k" of 0.125. Then one step of FOE was followed by 12 more steps of model search with stopping "k" set to .06. Finally one more step of FOE search placed the FOE at (41, 85) for the 128 meter initial model, and (43, 87) for the 64 meter initial model. The last step of the 32 meter initial model has two choices of equal minimum error, with the final FOE at (43, 87) or at (42, 85). The position (42,85) is believed to be the correct

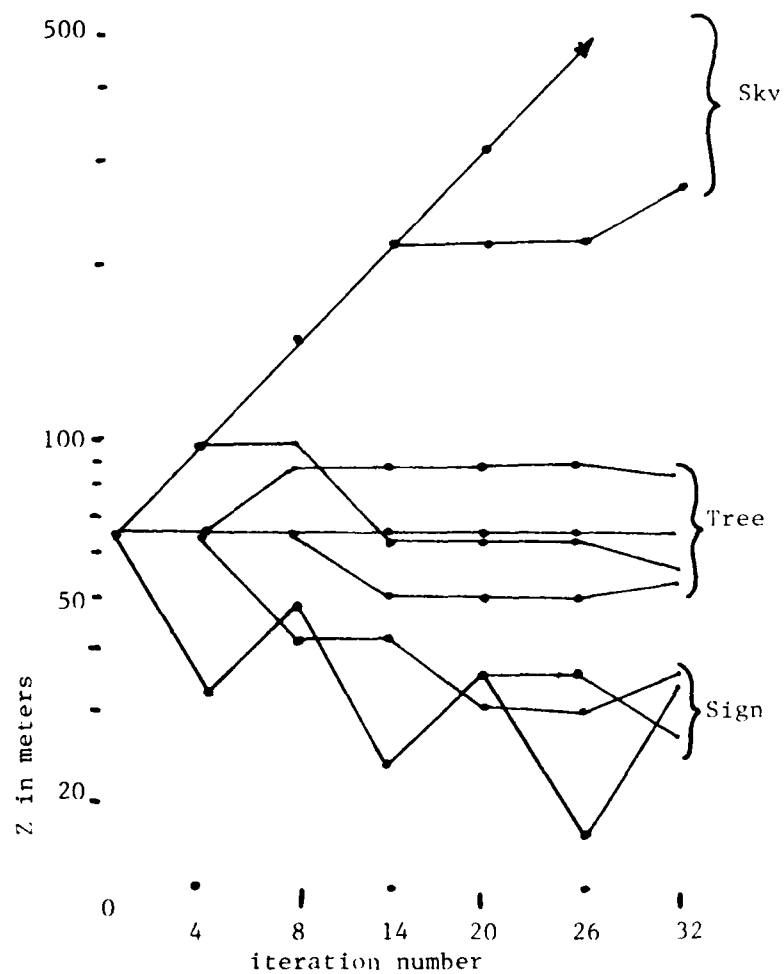


Figure 73. A sample of nine surfaces at the beginning, middle and end of each Z search. Iterations number 0, 8, and 20 had FOE search steps applied.



















Z SETTING METERS	DIRECTION RESULT FROM EACH ITERATION					
128						
64						
32						
ITERATION #	1	2	3 *	4 *	5 *	6

Table 4. Successive iterations of weighted FOE search run with various uniform settings of Z. A dot indicates that the central focus was chosen. An asterisk indicates the application of model search between those iterations of FOE search.



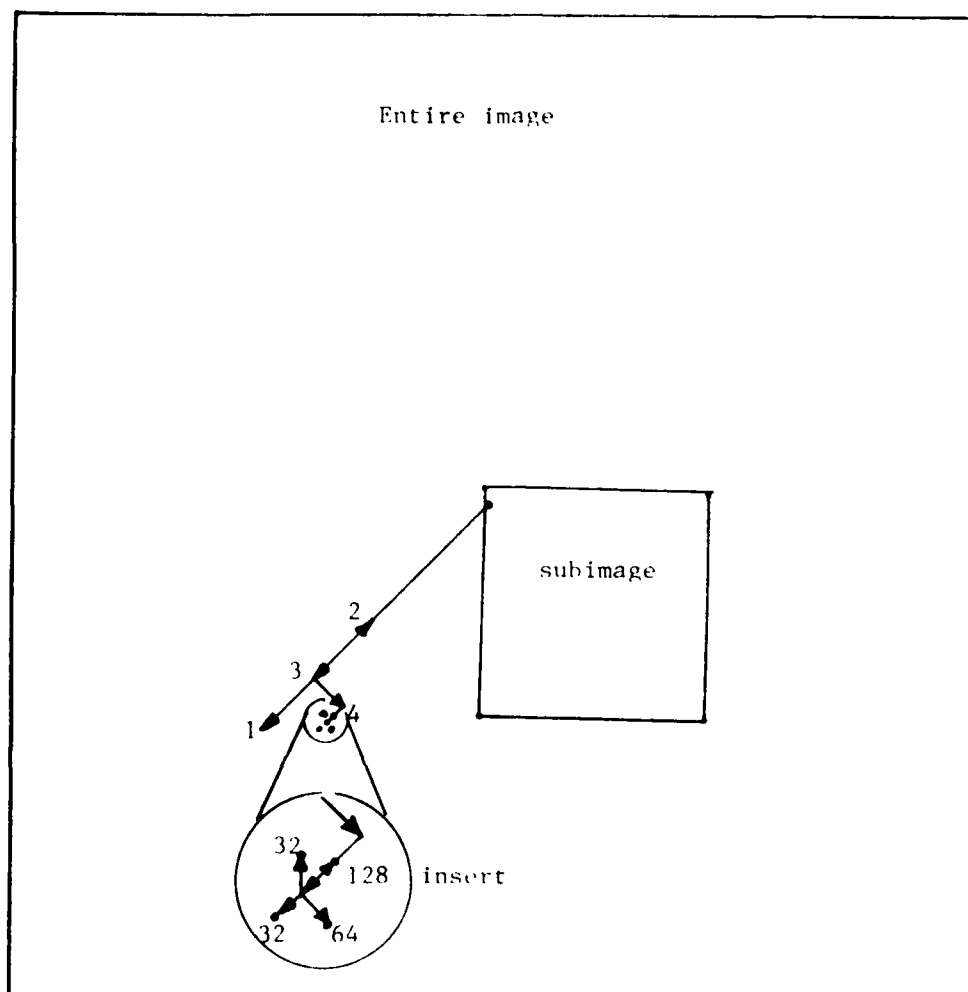


Figure 74. The six iterations of weighted FOF search. The insert indicates the final FOF placements for different initial models as discussed in the text.

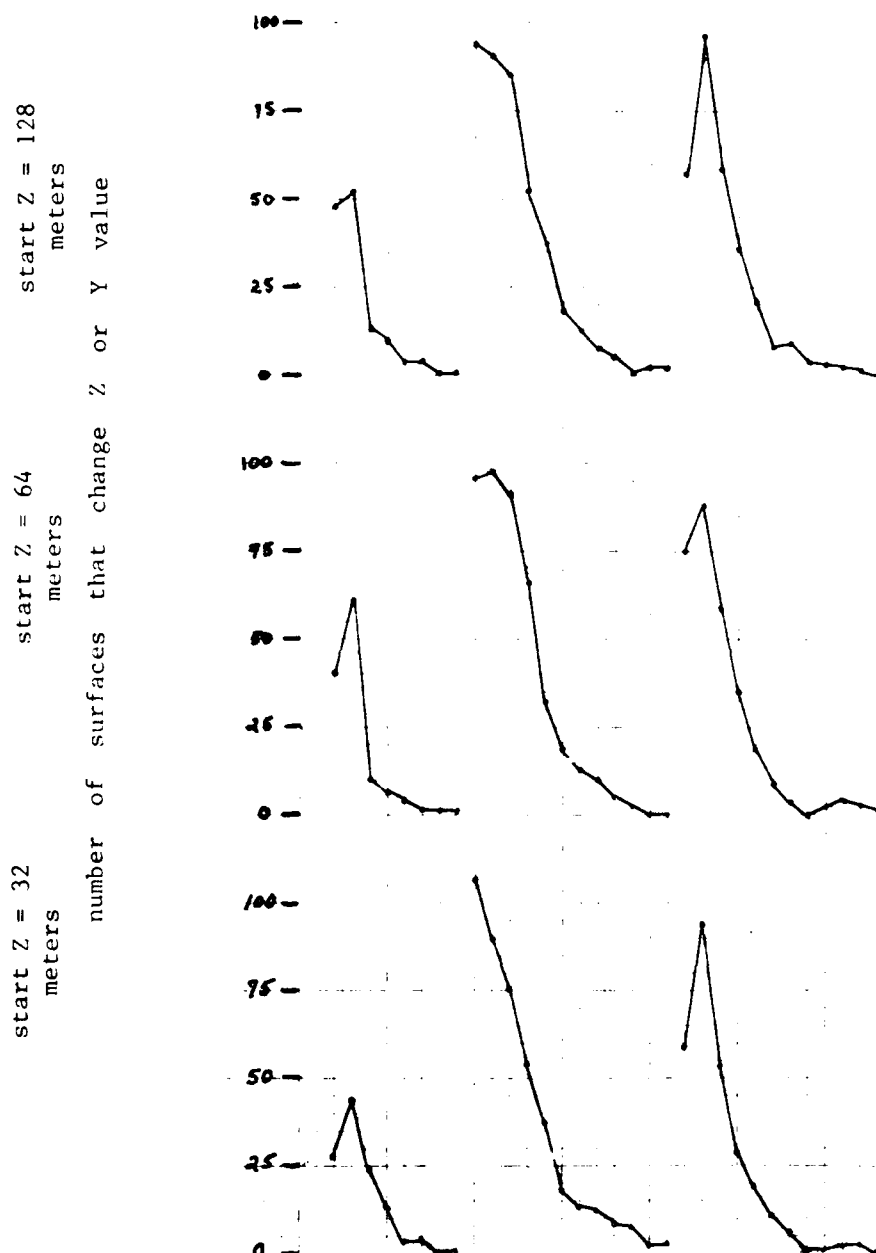
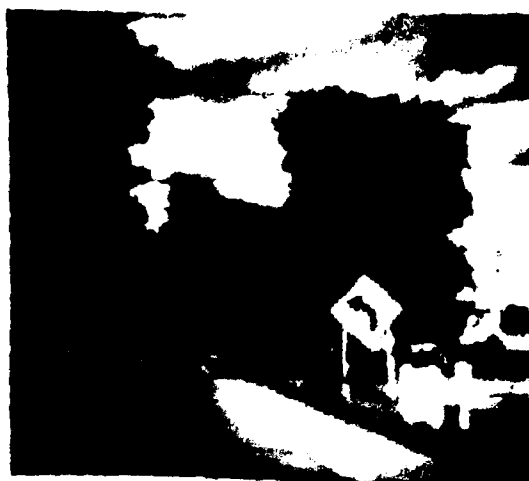


Figure 75. The number of surfaces that change value during iterations of Z and Y search. FOE searches are conducted between these searches.

placement of the FOE. Differences as small as two pixels have very little effect on the process or final model. This agreement in final FOE is considered a successful demonstration of immunity to initial distance assignment.

The decision of correct orientation for each surface is made on the basis of lowest error (see figure 76 for final model). Although two surfaces (on the road) were incorrectly posited as being vertical rather than horizontal we feel that the model is good. The two frames are only eight meters apart in space, and the horizontal regions contain very little visual texture.

IV.3.6 Experiment summary. The experiments presented here demonstrate the feasibility of constructing an interpretation from an image pair with very little human intervention. In some experiments, the automatically derived segmentation was then modified so that a nearly correct segmentation could be used to test the processes. Given such a segmentation, the processes behaved well. In the case where no human intervention occurred, the processes still performed quite effectively. The only major error was the incorrect decision on surface orientation for several surfaces.



a)



b)

Figure 76 a) the second image of pair #4 and b) the final surface model where distance is encoded as brightness.

The need to register the frames by hand was the result of small changes in camera orientation for which the cameraman and gyroscope did not compensate. It might be possible to automatically register the images by using correlation techniques, but this was not tried. One might also consider recording the inertial frame during filming and then using it for registration.

The use of weighted search for FOE makes possible the automatic generation of a surface model without first supplying a nearly correct surface model or FOE value. Thus, the system demonstrates (at least for one data set) a solution to the "start-up" problem. With a surface model constructed, the incorporation of it into a more complete interpretation of the images as objects will be the remaining problem considered in this thesis.

## CHAPTER V

### OBJECT INTERPRETATION

The identification of objects in natural images of outdoor scenes has been a research goal of image understanding for several years (Hanson 1978b, Bullock 1978, Tenenbaum 1976, Bajcsy 1974, Ohlander 1975, Levine 1978). The complexity of visual information and the difficulties involved in applying it to the problem of object identification led to the development of various representational structures. The representation presented here was developed to allow flexibility in the definition and application of various forms of visual knowledge for scene analysis (Hanson 1976 and 1978b, Williams 1977b). Demands for flexibility in the research environment has led to modularization of the representation. The resulting structure in the VISIONS system is a form of semantic network, where the knowledge is represented as interconnected graphs (Lowrance 1978).

Inference of surface and distance descriptions from static images has been a critical and challenging element in search of the goal of automated object identification. Once the spatial disposition of scene components is obtained, the inference of object identities becomes a

more tractable problem (Hanson 1978b, Marr 1977).

Consider an image with a centrally located green region which we assume to represent a surface. Basing an hypothesis solely on color, the region could represent a tree, grass, or perhaps a surface of an automobile. If the orientation of the surface(s) and its distance were known, the size of the object could be derived, and the set of possible identities could be narrowed to fewer (or even a single) object(s). Thus, the spatial disposition of surfaces is very useful for the identification of objects in general scenes.

Although several sources of visual knowledge have been explored within our representation, only two knowledge sources are examined here. One identifies objects based on size, and the other does so based on color and texture. To provide a framework to understand the application of this knowledge, this chapter begins with a section explaining the representation used, and then continues with a section about the knowledge implemented in the representation. The concluding section describes results obtained for object identification.

### V.1 Representation

Our representation of information in a static image is intended to describe the objects that appear in the scene being the imaged. This description is usually in terms of the objects' identity, their position in the image, their position in the world relative to the camera, and the spatial relationships between the objects. Similarly, a representation for objects in a dynamic image should include all of the aspects of a static representation, and additionally should characterize object dynamics, including the motions of objects, changes in identity, changes in appearance, observer dynamics, etc.

We have not addressed the problem of understanding objects with independent motion, and hence do not address the issues surrounding dynamic representations. For representation of motion concepts see (Badler 1976, Tsotsos 1976). Rather, the goal here is to infer surface properties from information gathered from a moving image. We have opted for a simple stationary representation of objects in the environment, frozen at time  $t_0$ . The depth properties, inferred from image dynamics, are used (along with other features) to then infer object



identity.

V.1.1 Levels of abstraction. The visual world can be described in terms of scenes that are composed of objects which satisfy a set of spatial relationships. The objects can be described in terms of their delimiting surfaces, and the surfaces have properties of color and textures, shape, size, position, etc. This descriptive hierarchy leads to a natural taxonomy of visual knowledge into abstraction levels (see figure 77), in a fashion similar to those used in speech understanding research (Erman 1975). Although one might posit more levels or perhaps a continuum of visual abstraction, the particular levels of surface, object and scene are probably essential to the object identification process (Williams 1977a, Parma 1980).

We are interested in a simple bottom-up approach to object identification, i.e. hypotheses based on image data. If the object interpretation system is given (or can determine) that the image is of a road scene, then the scene level of abstraction could be used to reduce the set of objects to be considered, and constrain their relative position. Although the discovery of scene identity can be a bottom-up process, the application of

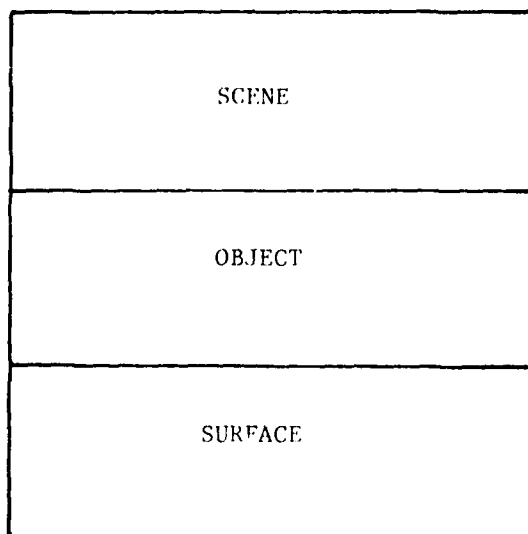


Figure 77. Abstraction levels for visual knowledge

scene information is essentially top-down. We will not discuss the application of knowledge at the scene level of abstraction. This thesis does not explore the possibilities of such processing, although it is clear that top-down processing is a key element in biological perception (Hochberg 1971, Price 1975, Arbib 1972, Spinelli 1967).

Bottom-up processing relies on as little a-priori knowledge as possible when inferring object identity. Features computed from the image data are matched with prototypes of features that are associated with stored object classes. Thus, although prototypical information is used in the identification process, the object identity is derived from information in the image, in a bottom-up fashion.

The result of the surface interpretation process is a model of the surfaces appearing in the images. This model contains the orientation and distance to each surface. Given this information, and the focal length of the camera lens, the three-dimensional size of the surface can be determined directly. Additionally, color and texture, measured over the areas of the image covered by the surfaces, is available for bottom-up analysis.

V.1.2 Short and long term knowledge. An interpretation can be viewed as a set of instantiated general concepts which must be related to the current environment. Thus, the representation within which interpretation occurs should be divided at every abstraction level into sections of short-term or image-specific, and long-term, or general information. The interpretation process acts to fill the short-term side of the representation and to relate it to both long-term concepts and image entities.

An example interpretation, where three objects are instantiated, is shown in figure 78. The scene level of abstraction is included in this example to show its place in the representation.

Nodes represent concepts such as object classes, or the orientations of surfaces. Arcs represent relationships between concepts such as super- and sub-class. This knowledge framework is therefore much like a semantic network (Quillian 1968, Woods 1975). but with partitions that collect substructures into loci of similar meaning and utility for the object interpretation process. These loci are called spaces (Hendrix 1975, Lowrance 1978). and can be viewed as separate graphs. Arcs connecting nodes in different spaces relate entities

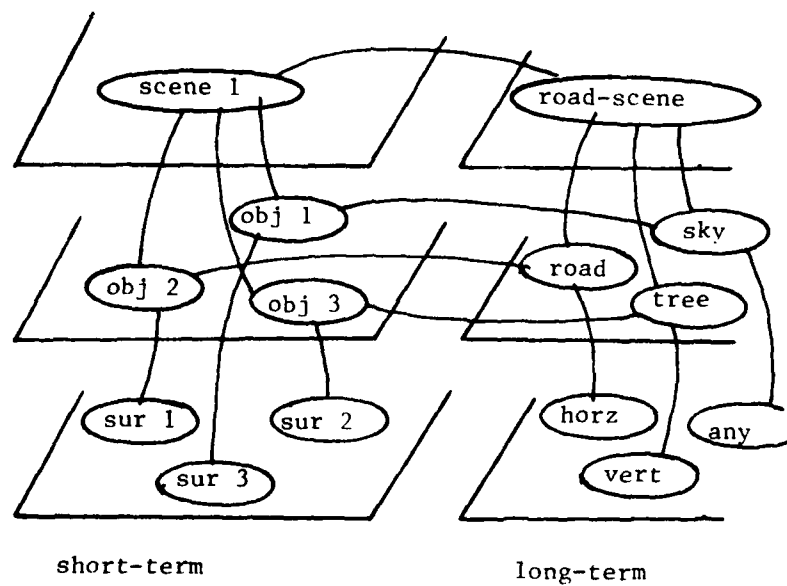


Figure 78. An example of an interpretation where three objects are instantiated

at different levels of abstraction, such as the surfaces that delineate an object, or the objects that participate in a scene. Arcs connecting nodes in short-term spaces with those in long-term spaces indicate instances of long-term concepts, such as an instance of the object class "TREE", or an instance of a scene class "ROAD-SCENE".

#### V.2 Knowledge Sources

The processes that act to form an interpretation are called Knowledge Sources or KSs (Erman 1975). Because the distinction between facts and hypotheses is maintained in the representation by division of the structure into long-term and short-term parts, it is clear where the KSs accept input and produce results. Additionally, because knowledge is divided into abstraction levels, the domain of each KS is usually restricted to a small set of spaces (often one) and the range of the KS is typically some other single space of the representation.

For example, consider a knowledge source that bases its hypotheses on the three-dimensional size of an object. The short-term surface space would contain surfaces as interpreted information derived from the

images, while the long-term surface space would contain surface size and orientation as expected for the general object classes which they describe. The size KS would then compare the derived size of imaged surfaces with the stored sizes in order to hypothesize object identities. One strategy is for the best match to be instantiated by placing a node in the short-term object space, and linking it to the long-term object-class node and the derived surface node via arcs across and down the knowledge structure as shown in figure 79. Examination of plausible strategies for organizing the application of knowledge sources and for selecting between possibly competing hypotheses is not the subject of this thesis, although selection of appropriate strategies is important in the interpretation process (Parma 1980).

Two KSs have been developed for object identification. One bases its result on the color and texture measured over the interpreted surfaces. The other is based on the three-dimensional size of an object derived from the distance and orientation of each surface. Both of these KSs apply an attribute matching technique that produces an heuristic confidence measure for the identity of each surface. The simple strategy employed for object identification is to apply both KSs

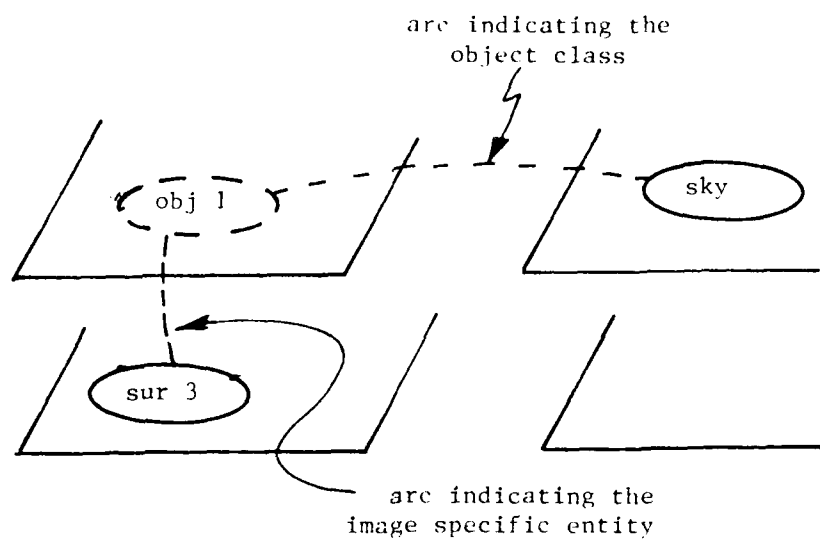


Figure 79. When an object is instantiated, a node is placed in the representation with arcs connecting it to the object class and the image entity.



to each surface, deriving for each one a confidence measure for all object classes in the data base. The highest resulting confidence indicates the object class that best identifies a surface. Adjacent surfaces with the same identification can then be joined into a single object.

Knowledge source development and application has proceeded through three phases. First, data were collected from images to form a data base of object attributes. Second, attribute prototypes for each object were abstracted from the data base. Third, a matching process was constructed to compare attributes extracted from the image with the stored prototypes to derive a confidence value.

The prototype and matching phases are very similar for the size and color KSs. The data base for the color KS was formed from digitized images, but the size KS prototypes were estimated and input by the author because the three-dimensional size of an object is available from our real-world experience.

V.2.1 Data collection. Data were collected by interactively selecting rectangular areas of images that each contained a single object class. From the eight images in the system library when the data were collected, 66 samples were selected. For each rectangular area the name of the object class was recorded and 11 features were computed. The average and standard deviation (termed "S.D." hereafter) of each feature were computed over the sample rectangle and stored in the data base. The two simple statistics were assumed sufficient to characterize each sample, even though some samples did not have normal distributions.

The data base formed from these samples consisted of averages and S.D.s of 11 features for each of 66 samples. The majority of the samples were taken from the object classes bush, grass, road, sky and tree. The proportion of object class samples are shown in figure 80. Samples other than those of the five target objects were taken in an attempt to reduce the chance of false matches on non-target objects.

The features measured were of two basic types. The first consists of eight point features, which are features that can be computed from the data at a single

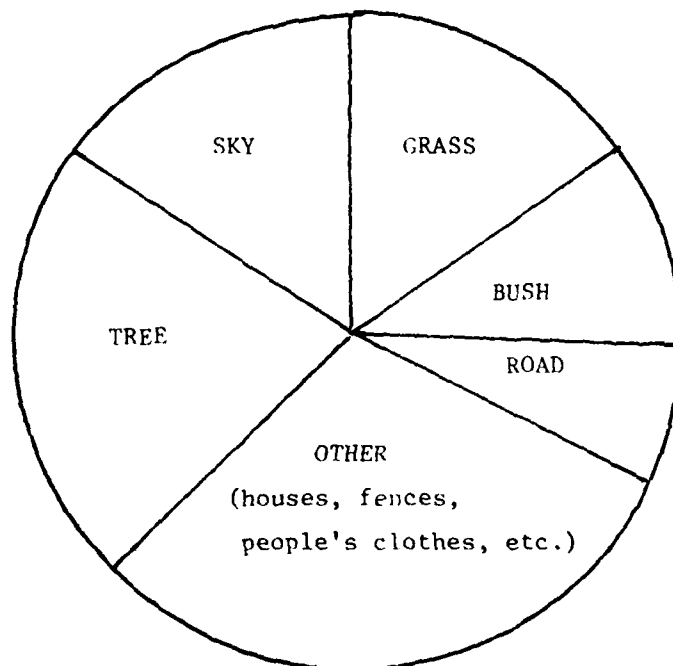


Figure 80. The data base consisted of 66 samples in proportion to this chart.

pixel. This includes the original red, green and blue data, intensity and various color transforms. The second consists of features computed over areas, and are loosely called "texture features". These features are measurements of the number of edges in two directions, and the average interpixel contrast.

V.2.2 Prototype formation. A prototype is abstracted from the data base for each object and each feature. Because of the variability of color and texture among man-made objects, such as automobiles, houses, and people's clothes, the set of objects classes was restricted to the five classes of bush, grass, road, sky, and tree.

Prototypes were formed as heuristic functions, and given weights which reflect the importance of each feature for each object class. The simplicity of this heuristic prototype allowed incorporation of estimates during the formation of size prototypes.

For each feature  $f_i$  of each object class  $O_j$  there is a range of possible feature values  $X_{ij}$ . Statistics that summarize the distributions of the means and the S.D.s of each feature value across the sample population of each object are used to form the prototypes. Thus,

for each object  $O_j$  a number of prototypes are formed, one for the mean of each feature, and one for the S.D. of each feature.

Consider the feature "raw green", across all samples of the object class "TREE" (figure 81). The average values of raw green across all data base samples of the object class "TREE" might range from 10 to 30, indicating that trees' greenness ranges from small to medium values. The S.D. values however, might all be nearly the same, say from 5.5 to 6.0. This would indicate that although the average tree greenness varies considerably over the samples, the variation of tree greenness varies little across the samples. To capture this important characteristic, a separate prototype was formed for the average and for the S.D. of each feature for each object class.

The ability of each feature to discriminate between the target objects is not the same. Consider the feature "raw blue" as a discriminator of the object class "SKY". In the data base very few samples, other than those for sky, have an average value of raw blue within the interval of the minimum and maximum average raw blue values for the samples of sky. Thus, average raw blue is

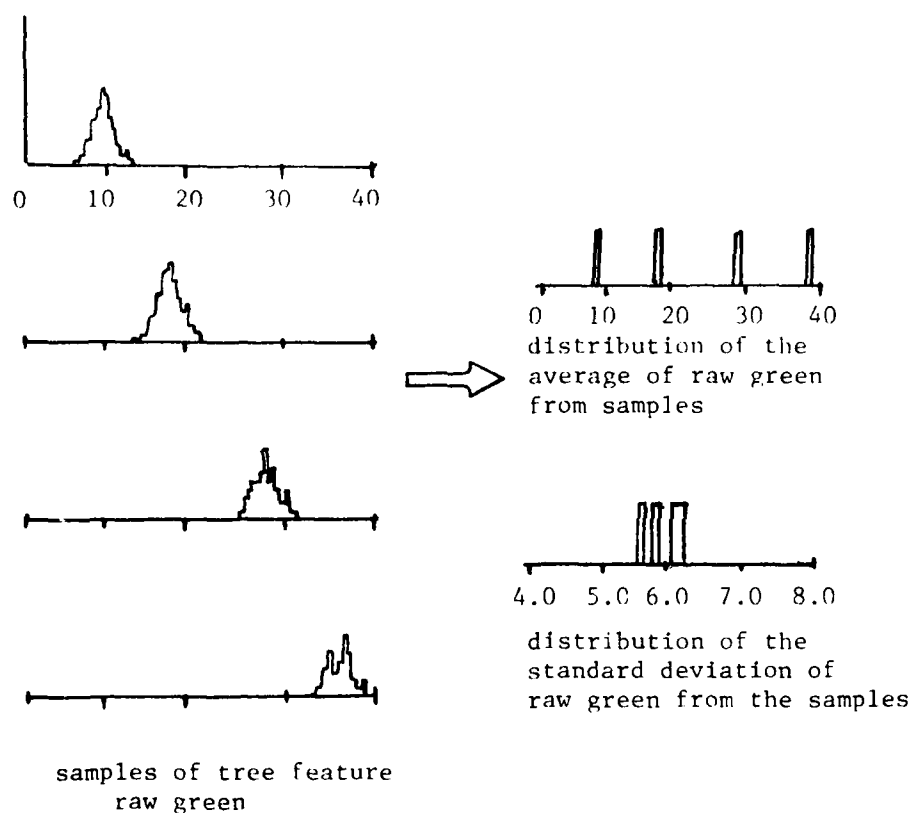


Figure 81. Both the average and the standard deviation are used as features in the data base. This example shows that the average can vary widely between samples, while the standard deviation remains the same.

a very good discriminator for the object class "SKY". On the other hand, the interval of minimum to maximum raw blue values for "TREE" almost completely overlaps the interval for samples of the object class "GRASS". Although raw blue is a good discriminator between the object class "SKY" and other object classes, it is not good for discriminating the object class "TREE" from the rest of the object classes.

Let us attempt to account for the ability of each feature  $f_i$  in discriminating object  $O_j$  from all other objects. One must remain aware of the fact that samples across images are subject to several sources of variability, such as lighting, color processing, digitization, etc. Because we have a small training set, and considerable variability, standard statistical pattern recognition approaches to determine each feature's power of discrimination were rejected. Instead, a weight  $w_{ij}$  is calculated by checking the number of data base samples that fall within an interval associated with a particular object. The interval of minimum to maximum  $x_{ij}$  was chosen because this interval includes all samples of the object and represents the limits of variation observed in the training set (data base). When all data base samples for  $O_m$  (where  $m$  is

not equal to  $j$ ) fall within the interval  $[\min(X_{ij}), \max(X_{ij})]$  the weight  $w_{ij}$  should be zero, indicating that feature  $f_i$  is useless for discriminating object  $O_j$  from others in the data base. When they are all outside the interval the weight  $w_{ij}$  should be one, indicating a perfect discriminating feature for  $O_j$ . The weight  $w_{ij}$  is calculated as a ratio:

$$w_{ij} = \frac{\# \text{ outside interval}}{\text{total \# samples } X_{im} \neq j} \quad \text{where } m$$

The numerator is the number of samples (excluding those for  $O_j$ ) for which the feature value falls outside the minimum to maximum interval of object  $O_j$ , and the denominator is the total number of samples (excluding those for  $O_j$ ).

V.2.3 Matching. The color and texture KS is designed to match the feature values of a surface in an image to the prototypes of each object class. The result of the match is a confidence value for the identity of the region as a particular object. The object that produces the maximum confidence value from the matches of one surface with a number of object class prototypes is then chosen to identify the surface.



The matcher uses a weighted average of match values obtained by comparing the surface's feature values with a function derived from the prototype values for the object. We express each match value as  $m_{ij}(X_{ik})$ . The notation  $X_{ik}$  refers to the  $f_i$  feature value for the surface  $k$ .

The prototype match function  $m_{ij}$  is constructed from the prototype values for the average, standard deviation, minimum and maximum of feature  $f_i$  for all samples of  $O_j$  that are in the data base. Note that when the feature is the average of raw blue, for instance, we compute the average, the S.D., the minimum and the maximum of the set of samples of average raw blue in the data base for object  $O_j$ . When the feature is the S.D. of raw blue, we compute the average, S.D., minimum and maximum of the S.D. of raw blue samples in the data base for object  $O_j$ .

Now we will form the heuristic prototype function  $m_{ij}$  shown in figure 82 with no assumptions about the statistical validity of its use. However it is being constructed to be robust within the (possibly wide) range of feature values that have appeared for each object in the training set. This function is flat topped, yielding

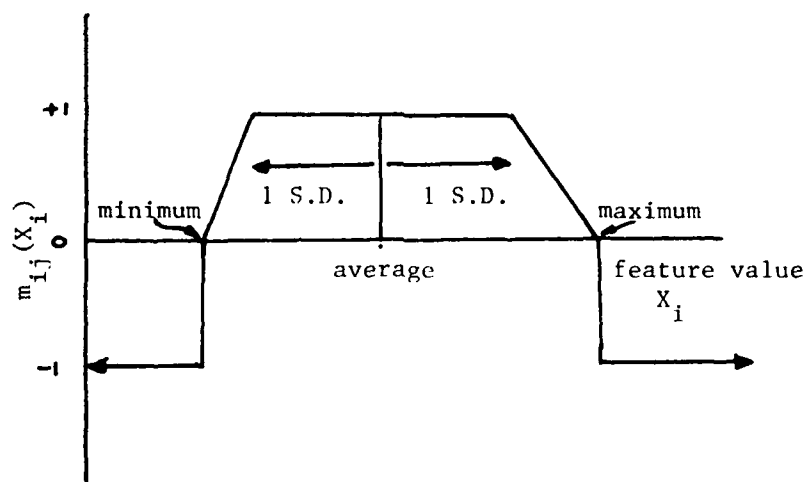


Figure 82. The matching function

a value of 1.0 for the interval of one standard deviation above and below the average. The function decreases linearly to zero as the feature value moves from one standard deviation above and below the mean to the max and min values, respectively; and the function takes on a value of -1.0 outside this range. The feature value  $X_{ik}$  for feature  $f_i$ , measured from the surface  $k$ , is the argument of the matching function, and the range of the function is  $\{-1\} \cup [0,1]$ .

Each feature value for a region results in one  $m$  value for each object. If the value  $X_{ik}$  is within a S.D. of the average expected value, the confidence is maximum, and therefore contributes strongly toward the hypothesis that the surface represents the object  $O_i$ . If it falls outside this range, but still within the minimum and maximum of data base entries, it results in less (but still positive) confidence. If it falls outside the range of values in the data base, the result is strongly against the hypothesis that the surface represents the object  $O_j$ , because no sample for  $O_j$  has been observed in this range.

The weighted average  $C_{jk}$  for one object  $O_j$  given a surface  $k$ , is formed from all feature matches  $m_{ij}(X_{ik})$ , and accounts for the feature discrimination power thus:

$$C_{jk} = \frac{\sum_i w_{ij}(X_{ik})}{\sum_i w_{ij}} .$$

The results for all  $O_j$ , given one surface  $k$ , can then be compared. The maximum of  $C_{jk}$  across all  $j$  is chosen for the hypothesis that object  $j$  is the identification for surface  $k$ . The resulting confidence ranges over the interval  $[-1,+1]$ , and is interpreted as evidence in support of the hypothesis if positive, and against the hypothesis if negative. Zero is interpreted as the point of no information.

V.2.4 Size prototype and matching. The size of three-dimensional bodies can be expressed in the dimensions of height, width and depth. This implies a standardized view, since a rotation of  $\pi/2$  radians about any axis switches two of the dimensions in a fixed coordinate system. The objects that we deal with have one fixed axis, which is the vertical, or  $Y$  axis. This is because our objects have functional dependence on gravity, and one particular axis is always aligned with the gravitational field. Our camera does not rotate appreciably with respect to the  $Y$  axis, and the  $Y$  axis

maintains good alignment with gravity. The Z and X axes are arbitrarily oriented with respect to any scene object. Therefore, the distinction between width and depth is difficult to make.

Fortunately, many objects in our scene have approximately the same dimensions of width and depth, or else there is an expected view of either width or depth. With these assumptions, we can define the prototype size of an object in two dimensions. The prototypes were generated ad hoc by the author. Figure 83 shows the prototype matching functions for height and width of our objects.

Size matching is performed in a manner similar to color matching. The two dimensions of size (where  $i=1$  or  $2$  represents height or width, respectively) are each considered a feature, and assigned a weight of 0.5 each. No weights are assigned the size features; they have equal influence on the size KS result. The average of the  $m_{ij}$  values then forms the resultant confidence value for the size KS.

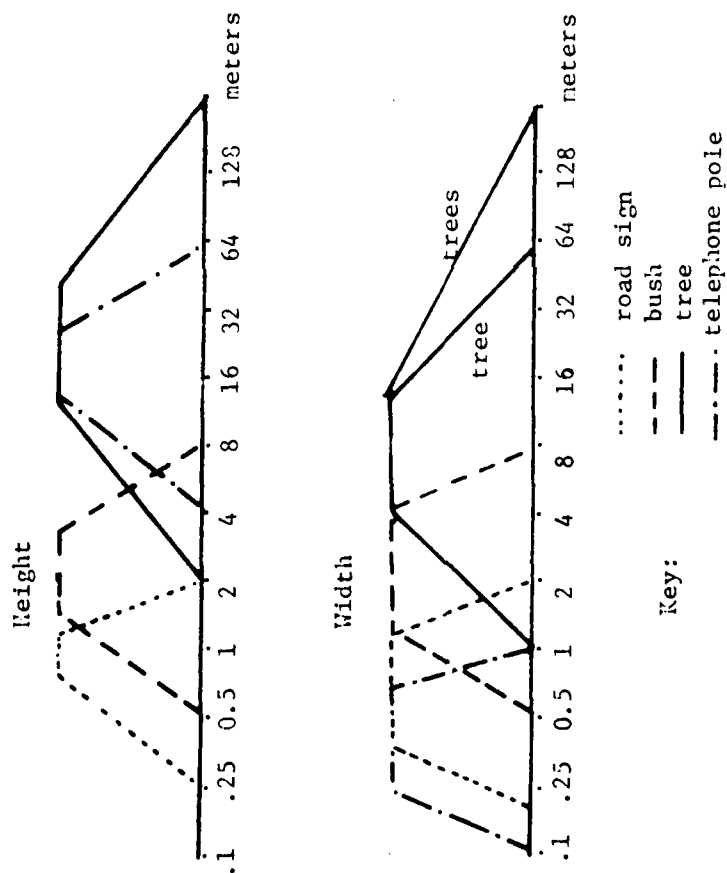


Figure 83. The matching functions for the size KS. The negative portions are not shown for clarity.

### V.3 Results

V.3.1 Results for the color KS. The color KS was tested on two static images to assess its performance. Table 5 summarizes these results. It was then run on the data from frame #45 and these results are presented at the end of this section, but are not part of table 5.

Because texture measures can only be computed in regions large enough to fit the texture computation window (a 5 x 4 area), one portion of the table removes the results for regions in which no texture windows fit. Also, to check the performance on larger regions, another category is presented for regions with areas greater than 65 pixels. The original object set consisted of bush, grass, road, sky, and tree. Because bush and tree have very similar characteristics, each category was divided into sub-categories. In one, the original five objects were used, and in the other, trees identified as bushes, and vice versa, were counted as correct identifications of the super-class foliage. Neither of the images contained a sample of road.

	ALL REGIONS		REGIONS IN WHICH AREA FEATURES WERE MEASURED		LARGE REGIONS (Greater than 65 pixels)	
	5	4	5	4	5	4
Number of objects	5	4	5	4	5	4
Number of regions	209		83		45	
Target regions	99		50		25	
Non-target regions	110		33		20	
Targets correct	76	91	40	45	19	23
Targets incorrect	23	8	10	5	6	2
% Targets correct	76.7	91.9	80.	90.	76.	92.

Table 5. Summary of region identification results of the color KS applied to two images of house scenes. The five object classes were bush, grass, road, sky and tree. The four class results combine bush and tree into a "foliage" class.



The distributions of the highest confidence values for each region are displayed in figure 84. These were compiled with the tree/bush distinction removed. They generally show that correctly identified targets have a higher confidence than the non-targets, although there still is significant overlap of confidence ranges. Not surprisingly, non-target surfaces can have spectral attributes very similar to the target objects. For example, in the images used here, the house roof is very similar to grass in the texture features, while the white wall is similar to sky in color features.

Several non-targets which resulted in large confidence values were examined. It was impossible to distinguish them visually from the associated target objects when the data were removed from the images and displayed next to each other (out of context). Additionally, the feature values were very similar. One particular case was a portion of a white house which was shaded from direct sunlight by a tree. The illumination was therefore skylight and appeared exactly like the sky in that image.

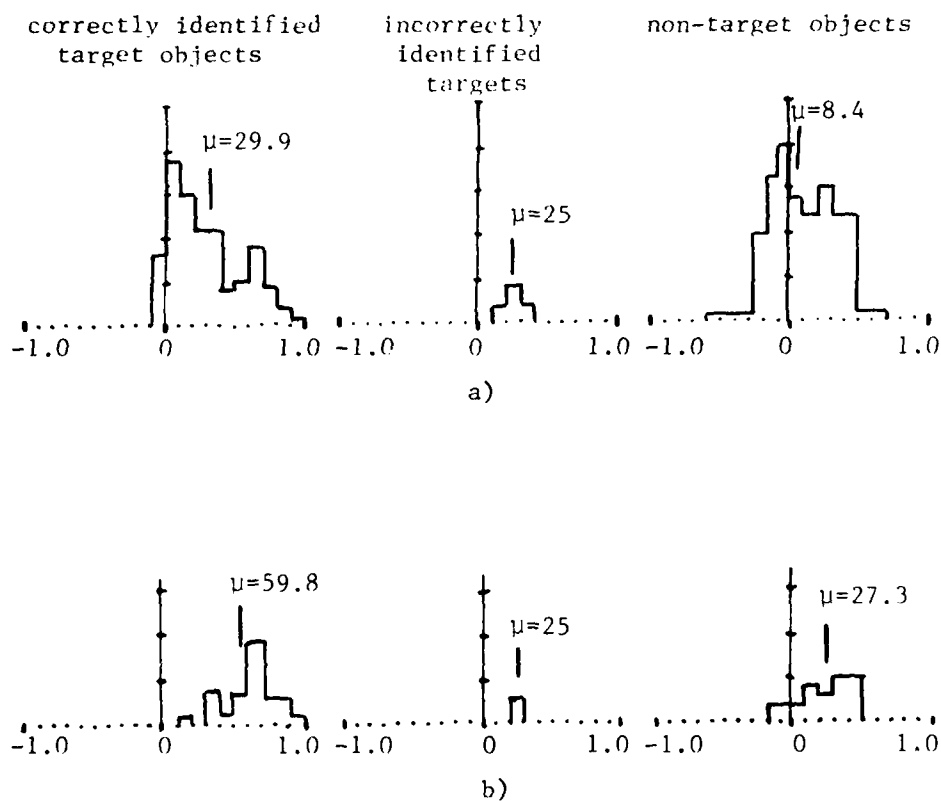


Figure 84. Distributions of confidence values for color KS applied to two static images, where a) shows four object classes and all regions, and b) shows four object classes and regions with area greater than 65 pixels.

Table 6 shows the results from the color KS applied to frame #45. The segmentation is shown in figure 85. The major surfaces are presented here (see appendix for complete listing). The three selected sky surfaces and two selected tree surfaces are correctly identified. Surfaces 48 (telephone pole) and 86 (sign) are included because they were selected (by hand) for size analysis. Surfaces 131 and 153 are entirely road, and identified as sky. A close examination of the data and the images reveals that the road appears exactly like portions of the sky in this image. Therefore, if we assume that the road and sky can look similar, we should expect that horizontal objects with sky color which appear below the horizon to be road. Surfaces 109 and 125 are a combination of road and grass. Although 125 was identified very slightly as grass, it contained very little - it was mostly road. Number 109, which was mostly grass was identified as sky or road. The grass in these images was not very green, and since the scene was imaged in Autumn, and there were not very many representative samples of this type of foliage, one can expect some error here that might be removed with a more comprehensive data base.

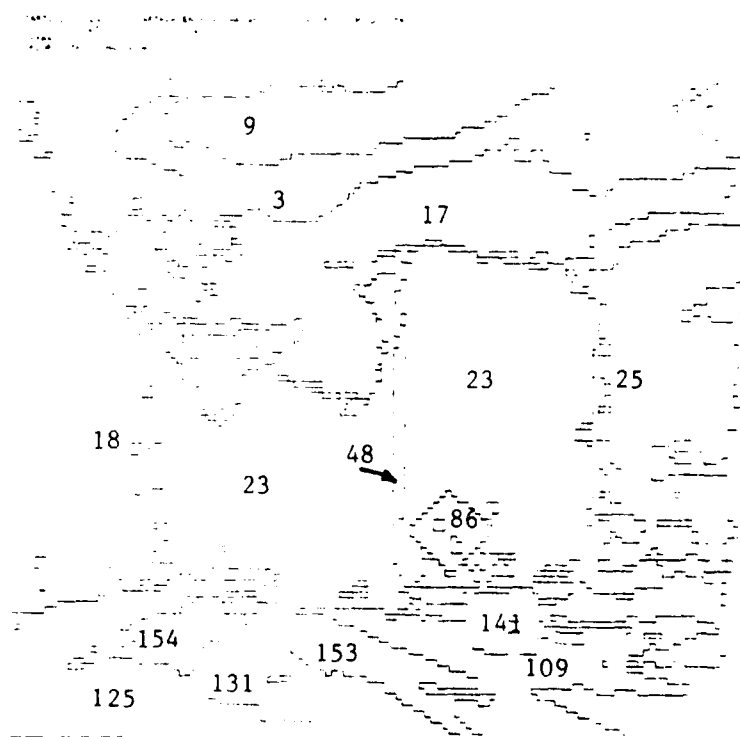


Figure 85. Segmentation of averaged image  
with surfaces labeled by number

SURFACE NUMBER	OBJ 1 BUSH	OBJ 2 GRASS	OBJ 3 ROAD	OBJ 4 SKY	OBJ 5 TREE	ACTUAL IDENTITY	KS HYP	FILTERED HYP
3	-.72	-.84	.41	.74	-.98	SKY	SKY	SKY
9	-.91	-1.00	.08	.88	-.98	SKY	SKY	SKY
17	-.69	-.60	.48	.91	-.88	SKY	SKY	SKY
18	-.06	-.86	-.83	-.92	.20	TREE	TREE	TREE
23	-.42	-.32	-.60	-.42	.44	TREE	TREE	TREE
48	-.54	-.49	.30	.01	-.87	T'POLE	ROAD	SKY
86	-.11	-.09	-.05	.66	-.48	SIGN	SKY	SKY
109	-.74	-.67	.47	.92	-.91	ROADSIDE	SKY	ROAD
125	-.27	.03	-.46	-.27	-.32	ROADSIDE	GRASS	GRASS
131	-.63	-.59	.54	.90	-.89	ROAD	SKY	ROAD
141	-.21	-.29	-.43	.73	-.29	GUARDRAIL	SKY	ROAD
153	-.13	-.02	-.54	.99	-.23	ROAD	SKY	ROAD

Table 6. Results of the color KS on selected surfaces.  
 Sky and tree were identified correctly. Surfaces numbered  
 109 and 125 are a mixture of road and grass. The use of  
 the filter is described in the text.

Except for the non-target problem, the color KS performed well. With only three images to judge by, one cannot quantify the results in a reliable manner. The incorrect results on frame #45 probably result from the facts that the road was reflecting a considerable amount of skylight and the grass was dully colored. The addition of a "horizon filter", where a sky interpretation below the horizon can be re-interpreted as road, and vice versa, corrects for the road problem. An alternative filter strategy is to just remove the sky choice and take the next largest (positive) confidence.

The approach, appears promising as a method for incorporating relatively few samples, or human supplied estimates into a system of knowledge application. Improvements might be made by utilization of larger training sets, and application of rigorous probability and decision theory. Additionally, the use of color standards, photographed along with the scene, could make normalization of the data possible.

V.3.2 Results for size matching. The size KS was tested on static images (with assumptions of distance given) (Hanson 1978b). The results using the surface interpretation for the image pair 4 is given in table 7. All objects (of the five tested) gave reasonable results. The trees and telephone pole were correctly identified, but the size of the sign (surface 86) fell within the prototype for bush, and matched that prototype better than the sign did. Since bush is in the target set for the color KS, this is a good candidate for improved recognition via a combination of the two KSs. The guard rail posts (surface 141) closely matched the prototype for sign also.

V.3.3 Combination of spectral attribute and size results. The color and size KS results were multiplied together to form a combined identity confidence. The maximum combined confidence value served to choose the object class for each hypothesized surface. The only objects hypothesized entirely by size are the roadsign and telephone pole. The only ones for which there is no size information are the sky, grass and road.

The combination of diverse sources of knowledge has been treated in diverse ways in different artificial intelligence research efforts (Erman 1975). Even in pattern recognition literature a number of combination algorithms have been presented, each with its own rationale for the combining function.

The choice of multiplication as the combining function coincides with the observation that size and color are independent sources of information. In the pattern recognition literature, the assumption of independence of features leads to a multiplicative combination with the maximum likelihood choice minimizing error.

To perform this multiplication, the confidence values are first put into the range interval  $[0,1]$ . Those objects for which no prototypes exist are assigned a value of 0.5. Thus, strong evidence against and hypothesis becomes represented by a 0., strong evidence in favor by a 1., and 0.5 becomes the no information point. Note that after combination (multiplication) the resulting value takes on a different interpretation. The combination of two 0.5 confidences becomes 0.25. Once values are multiplied the results cannot be judged as



SURFACE #	Z	HEIGHT		WIDTH	
		pixels	meters	pixels	meters
18	92	83	19.8	27	6.5
23	61	70	11.2	95	15.1
48	53	49	6.8	2	.27
86	35	14	1.3	14	1.3
141	24	11	.7	21	1.3

a)

SURFACE #	SIGN	BUSH	TREE	T'POLE	IDENTITY	HYPOTHESIS
18	-1.	-0.25	1.	0.	TREE	TREE
23	-1	-1.	1.	0.	TREE	TREE
48	-0.25	-0.95	-0.15	0.75	T'POLE	T'POLE
86	0.9	0.95	-0.25	-0.45	SIGN	BUSH
141	1.0	0.5	-1.	-0.25	GUARDRAIL	SIGN

Table 7. a) Size measurements for five surfaces in the image pair, as determined from the surface interpretation, and b) the results from matching these sizes with the object size prototypes.

having a reliable no-information point, but rather can only be compared to one another and the maximum chosen as the maximum likelihood class. The results for multiplicative combination are given in table 8. Figure 86 shows the interpretation in terms of objects.

SURFACE	SIGN	BUSH	GRASS	ROAD	SKY	TREE	T'POLE	ACTUAL IDENTITY
18	0.	.25	.03	.10	.14	.60	0.	TREE
23	0.	0.	.17	.10	.12	.72	0.	TREE
48	.17	.01	.12	.32	.27	.03	.38	T'POLE
86	.45	.23	.23	.24	.41	.09	.13	SIGN

Table 8. Results of multiplying the results of the size KS on the four selected regions by the results from the color KS. The results of the KSs were put in to the range [0,1] prior to multiplying. All of these major regions were correctly identified.



Figure 86. The object interpretation. Key: from brightest to darkest: Tree, Telephone pole, Sky, Sign, Road, Unclassified surfaces are black. All non-black surfaces are correctly identified, except for the guardrail and grass.

## C H A P T E R V I

### CONCLUSIONS

#### VI.1 Summary

The goal of this thesis is the implementation of a computer program that constructs an interpretation of moving images of a natural scene. Given movie frames as input, the system develops an internal representation of the scene in two stages. First, image dynamics are used to form a simplified surface model. Then, a model of the scene in terms of objects is derived from the surface model and other features of the images.

The following is a summary of specific accomplishments:

1. Development of a method for interpretation of a movie from a moving camera in a naturally occurring scene.
2. Development of a means for determining surface distance from moving images using a predictive model and point features.
3. The use of an hypothesize-test strategy.

4. A method for representing and using visual knowledge to build an object interpretation.

The focus of this thesis has been the development of mechanisms for the analysis of motion in order to derive surface hypotheses. The first portion of the system has been explored in detail since it is the basis for a surface interpretation of the physical environment. A model is hypothesized depicting the three-dimensional positions of scene surfaces relative to the camera. The motion of image features is used to refine this model as well as update it. The surface interpretation provides both the object/background segmentation and size measurement necessary for object identification. An object interpretation is produced by comparing both spectral features from the image and size information from the surface interpretation to features associated with stored object names.

A surface interpretation is first hypothesized, based on a coarse static analysis. Then the hypothesized three-dimensional model of scene surfaces is used to predict image dynamics which are tested through inter-image comparisons. This is in contrast to the more common motion detection techniques that detect image

dynamics. Such systems then must compose a three-dimensional model from the point velocities of particular trackable points or areas in the images. It is not easy to construct a three-dimensional model of the scene from the typically scattered set of point velocities that such systems produce. Often, because they rely on edge features, or correlation windows, (which are the image of borders between surfaces) such motion detection systems will give false indications of image motion because they may erroneously track portions of an image that belong to several surfaces. The problem is avoided in the system presented in this thesis by using pixel features and a predictive model.

This approach avoids the problems that other systems have in dealing with occlusion. With point features, occlusion does not give rise to false matches, and with a predictive model, the effect of occlusion is predicted.

Within our definitions, the interpretation of natural outdoor images culminates in an understanding of the spatial layout and identity of objects. Information on size, color, and texture is shown to be sufficient for the identification of objects in our scenes. Although the system has only been tested on one sequence of motion

images of a road scene, we believe that the surface interpretation process derives a surface model that would be sufficient for object interpretation in other domains.

### VI.2 Sources of Error

IV.2.1 Bad segmentation. Although re-segmentation can compensate for bad segmentations, it is no substitute for a good original segmentation. The image differencing technique used for re-segmentation can only extract the leading and trailing edges of areas that should be segmented when the areas are homogeneous. This is because image differencing shows those portions of an image pair that are different, and an homogeneous moving region will create differences with a size equal to its displacement. If this displacement is less than the size of the region (in the direction of motion) then only the leading and trailing edges will be detected by differencing.

Also, it is possible to achieve better initial segmentations through a "local" segmentation process (Nagin 1979). However, partitioning of the image creates problems with inter-image differencing around the edges of each partition. Localization of the feature histogram process leads to different cluster centers in each



partition of the image. Because different cluster centers are used, the feature vectors cannot be compared when the inter-image displacements cross partition borders. Therefore, the application of localized segmentation to motion is left for future work.

IV.2.2 Blur. Blur is caused by movement during the time when the shutter is open. Our camera has a 100 degree shutter, which translates to .015 secs, or .25 meters of travel, during the exposure of each frame. Thus, it is unreasonable to expect the system to resolve distances to this resolution. The effect of blur is to soften edges that lie perpendicular to an expansion line (radial line from the FOE) and to strengthen those edges that are colinear with expansion lines. We do not know if blur hinders or helps the refinement process. Consider the effect of blur on a sharply contrasting boarder between surfaces. It will give rise to a gradient in both the first and second images of a pair. It might be possible that such gradients will improve the refinement process for homogeneous surfaces because inter-image matches will have a variety of values when near to a correct match (thereby giving a smoothness to the error function). However, it is also possible that heterogeneous surfaces would do poorly because texture elements would be also

blurred and reduce the otherwise easily matched textures.

IV.2.3 Resolution. The effect of image resolution on refinement is not straightforward. Since the system interpolates to achieve sub-pixel resolution, it is not clear how much improvement would be gained by increasing spatial resolution. Increased resolution also means more image points, and thus, the computational expense increases.

Spatial resolution is related to distinguishability of texture elements in the scene. When the resolution is sufficient to distinguish visual texture, then the texture elements become sources of information for inter-image comparison. When the resolution is insufficient, two things can happen. Either the object appears homogeneous in both images, or the texture elements appear in one image but not the other. A homogeneous field can be tracked, but not as well as one with texture. Texture change between frames is due to the sampling process that forms the quantized image. Also, changes in lighting, orientation and movement, such as the leaves in trees, can change texture. These effects are somewhat compensated for by the interpolation process used during image synthesis.

The interpolation process itself is not optimum. One more suitable process would be a convolution of a gaussian with the predicted displaced position. Such a convolution kernel would require about 18 points and considerable computational expense. No kernel except the four point interpolater as described in chapter III was tried. Because the images contained no very sharp edges, it was felt that the simpler kernel was acceptable.

VI.2.4 Surface orientation. Some of the horizontal surfaces were incorrectly labeled as vertical surfaces. One possible cause for this error is the lack of texture on the road. Another is the lack of sharply contrasting regions on the road which did exist on other surfaces such as the sign. More work should be done to examine exactly what causes the residual error once the search has completed, and under what circumstances the lowest error is found for the correct orientation.

### VI.3 Suggested Future Improvements

VI.3.1 Better data. The dynamic range and resolution of the data were poor. This is the result of using super-8 film, and digitizing it through a television camera. Larger format movie film would have required a larger gyroscope for the more massive camera. Quality film

digitizers take considerable time to digitize each frame, and the use of such equipment was beyond the budget of this work. The fine segmentations that result from the use of wide dynamic range (and high resolution) images indicates that higher quality data could improve our process.

#### VI.3.2 Third orthogonal plane.

In other domains the third plane (XZ), and perhaps others, might be necessary. Their inclusion merely requires that a function be expressed which generates a Z for any point on the surface. The inter-image displacement can then be computed for each point, and refined using the Z refinement technique.

VI.3.3 Automatic foveation (and feedback). To produce accurate distance measurements and obtain detailed shape information for more sophisticated object recognition, the idea of a high resolution, steerable window, like the fovea of the human eye, becomes an attractive mechanism. Although such a mechanism was considered, it was not implemented.

#### VI.3.4 Implementation of dynamic object representation.

A dynamic object representation was designed, but not implemented, because it was not needed to achieve the primary goals of this thesis. This representation would extend the two-dimensional representation with the inclusion of time. The resulting three axes of knowledge in the representation would be short-term vs. long-term, abstraction, and time. Since surfaces change more rapidly than objects, and objects more than scenes, the representation would need fewer spaces at higher abstraction levels.

VI.3.5 More objects. More objects ought to be included, and far more samples should be taken to make the database of object attributes statistically meaningful. The design of the object interpretation section has no inherent limit on the number of objects. The computational cost increases linearly with the number of features, regions, and objects. Classification accuracy will likely decrease with a large number of objects, and with several more objects one would implement a KS where color and texture only provide constraints on identity rather than providing unique solutions. Therefore, the need for the scene level of description, which would restrict the number of objects to an appropriate few, would be highly desirable.

## B I B L I O G R A P H Y

- Arbib, M., The Metaphorical Brain, Wiley-Interscience, New York, 1972.
- Attneave, F., "Some Informational Aspects of Visual Perception," Psychological Review, vol 61, 1954.
- Badler, N., "Conceptual Description of Moving Objects," Ph.D. Dissertation, University of Toronto, 1976.
- Bajcsy, R. and L. Lieberman, "Computer Deescription of Real Outdoor Scenes," Proceedings 2nd International Joint Conference on Pattern Recognition, August 1974.
- Barrow, H. and J. Tenenbaum, "Representation and Use of Knowledge in Vision," Tech Note 108, Stanford Research International, Menlo Park CA, July 1975.
- Barrow, H. and J. Tennenbaum, "Recovering Intrinsic Scene Characteristics from Images", Tech Note 157, Stanford Research International, April 78
- Bullock, B., "The Performance of Edge Operators on Images with Texture," Technical Report, Hughes Research Laboratories, Malibu CA, Oct 1974.
- Bullock, B., "Unstructured Control Concepts in Scene analysis," Report 497, Hughes Research Laboratory, Malibu CA, June 1976.
- Bullock, B., et. al., "Finding Structure in Outdoor Scenes," Report 498, Hughes Research Laboratory, Malibu CA, July 1976.
- Bullock, B., "Unstructured Control and Communication Processes in Real World Scene Analysis," Report CS-1, Hughes Research Laboratory, Malibu CA, August 1977.
- Bullock, B., "The Necissity for a Theory of Specialized Vision," Computer Vision Systems, Academic Press, 1978.

- Burt, P., "Stimulus Organization Processes in Stereopsis and Motion Perception," Ph.D. Dissertation, Tech Report 76-15, Department of Computer and Information Science, University of Massachusetts, Amherst, MA, Sept 1976.
- Cloksin, W., "Perception of Surface Slant and Edge Labels from Optical Flow: a Computational Approach," Paper #33, Department of Artificial Intelligence, University of Edinburgh, Edinburgh Scotland, 1978.
- Dreschler, L. and H. Nagel, "Using Affinity for Extracting Images of Moving Objects from T.V.-Frame Sequences," Tech Report 44-78, Department of Information Science, University of Hamburg, Hamburg Germany, February 1978.
- Duda, R., D. Nitzan and P. Barrett, "Use of Range and Reflectance Data to Find Planar Surface Regions," IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol PAMI-1, July 1979.
- Erman, L. and V. Lesser, "A Multi-level Organization for Problem Solving Using Many, Diverse, Cooperating Sources of Knowledge," Tech Report, Department of Computer Science, Carnegie-Mellon University, Pittsburg PA, 1975.
- Freuder, E., "Active Knowledge," Vision Flash 53, AI lab MIT, Cambridge MA, October 1973.
- Garvey, T., "Perceptual Strategies for Purposive Vision," SRI Project 3805 Tech Note, Stanford Research International, Menlo Park CA, September 1976.
- Gibson, J., The Perception of the Visual World, Greenwood Press, Westport CT, 1950.
- Gibson, J., The Senses Considered as Perceptual Systems, Houghton-Mifflin, Boston MA, 1966.
- Hannah, M., "Computer Matching of Areas in Stereo Images," Ph.D. Thesis (Report AIM-239), Department of Computer Science, Stanford University, Stanford CA, 1974.

- Hanson, A. and E. Riseman, "The Design of a Semantically Directed Vision Processor (Revised and Updated)," Tech Report 75 c-1, Department of Computer and Information Science, University of Massachusetts, Amherst MA, February 1975.
- Hanson, A., E. Riseman and T. Williams, "Constructing Semantic Models in the Visual Analysis of Scenes," Proceedings of the IEEE Milwaukee Symposium on Automatic Computation and Control, p97-102, April 1976.
- Hanson, A. and E. Riseman, "Segmentation of Natural Scenes," from Hanson and Riseman (ed.), Computer Vision Systems, Academic Press, 1978.
- Hanson, A., et. al., "VISIONS: A Computer System for Interpreting Scenes," from Hanson and Riseman (ed.), Computer Vision Systems, Academic Press, 78.
- Hendrix, G., "Partitioned Networks for the Mathematical Modeling of Natural Language Semantics," Tech Report NL-28, Ph.D. Thesis, Department of Computer Science, University of Texas at Austin, Austin TX, December 1975.
- Hochberg, J., Woodsworth and Schlosberg's Experimental Psychology, 3rd Ed., Holt, Rinehardt and Winston, pp 395-473, 1971.
- Horn, B., "Shape from Shading: a Method for Obtaining the Shape of a Smooth Opaque Object from One View," MAC Tech Report 79, MIT, Cambridge MA, 1970.
- Jain, R. and H. Nagel, "On a Motion Analysis Process for Image Sequences from Real World Scenes," Tech Report 48-78, Department of Information Science, University of Hamburg, Hamburg Germany, 1978.
- Lee, D., "Visual Information During Locomotion," from Macleod and Pick (Ed.), Perception: Essays in Honor of J.J. Gibson, Cornell University Press, Ithaca NY, 1974.
- Levine, M., "A Knowledge Based Computer Vision System," from Hanson and Riseman (Ed.), Computer Vision Systems, Academic Press, 1978.



- Limb, J., and J. Murphy, "Estimating the Velocity of Moving Images in Television Signals," Computer Graphics and Image Processing, Vol 4., December 1975.
- Lowrance, J., "Grasper 1.0 Reference Manual," Tech Report 78-20, Department of Computer and Information Science, University of Massachusetts, Amherst MA, December 1978.
- Luria, A., The Working Brain, Basic Books, 1973.
- Marr, D., "Representing Visual Information," Tech Report AIM-415, AI lab MIT, Cambridge MA, MAY 1977.
- Martin, W. and J. Aggarwal, "Dynamic Scene Analysis: The Study of Moving Images," Tech Report 184, Information Systems Research Laboratory, Univ of Texas at Austin, January 77.
- Milgram, D., "Region Tracking Using Dynamic Programming," Tech Report TR-539, Computer Science Dept, University of Maryland, College Park MD, May 1977.
- Nevatia, R., "Depth Measurement by Motion Stereo," Computer Graphics and Image Processing, May 76.
- Nagin, P., "Computer Segmentation of Natural Scenes," Ph.D. Thesis, University of Massachusetts, Amherst MA, 1979.
- Parma, C., A. Hanson and E. Riseman, "Experiments in Schema-Driven Interpretation of a Natural Scene," in Proc. of the Workshop on Picture Data Description and Management, IEEE, August 1980.
- Palander, R., "Analysis of Natural Scenes," Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh PA, April 1975.
- Piliphuck, A., personal communication, University of Maryland, 1977.
- Prager, J., "Analysis of Static and Dynamic Scenes," Ph.D. Thesis, University of Massachusetts, Amherst MA, 1979.

Price, K., "A Comparison of Human and Computer Vision Systems," Association for Computing Machinery SIGART Newsletter, No. 5, February 1975.

Price, K., "Change Detection and Analysis in Multi-Spectral Images," Ph.D. Thesis, Carnegie-Mellon University, Pittsburg PA, 1976.

Quam, L., "Computer Comparison of Pictures," SAIL Memo AIM-144, Computer Science Dept, Stanford University, Stanford CA, May 1971.

Quam, L. and M. Hannan, "Stanford Automatic Photogrammetry Research," SAIL Memo AIM-254, Computer Science Dept, Stanford University, Stanford CA, December 1974.

Quillian, R., "Semantic Memory," from Minsky (ED.), Semantic Information Processing, MIT Press, Cambridge MA, 1968.

Radig, B., "Description of Moving Objects Based on Parameterized Region Extraction," Tech Report 61-78, Department of Information Science, University of Hamburg, Hamburg Germany, April 1978.

Riseman, E. and M. Arbib, "Computational Techniques in the Visual Segmentation of Static Scenes," Computer Graphics and Image Processing, Vol. 6, 1977.

Riseman, E., private communication, University of Massachusetts, Amherst, 1980.

Roach, J. and J. Aggarwal, "Computer Tracking of Objects Moving in Space," from IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol PAMI-1, #2, April 1979.

Roberts, L., "Machine Perception of Three-Dimensional Solids," Ph.D. Thesis, MIT, Cambridge MA, 1965.

Rosenthal, D. and R. Bajcsy, "Conceptual and Visual Focussing in the Recognition Process as Induced by Queries," Proc. 4th International Conference on Pattern Recognition, 1978.

Rosenfeld, A., Talk given at the IEEE sponsored conference "Workshop on Computer Analysis of Time-Varying Imagery", Philadelphia PA, April 1979.

Shirai, Y., "A Context Sensitive Line Finder for Recognition of Polyhedra," Artificial Intelligence, Vol. 4, Summer 1973.

Simon, H., The Sciences of the Artificial, MIT press, Cambridge MA, 1969.

Spinelli, D., and K. Pribram, "Changes in Visual Recovery Function and Unit Activity Produced by Frontal and Temporal Cortex Stimulation," Electroenceph. and Clinical Neurophysiology, Vol 22, pp 143-9, 1967.

Tenenbaum, J., "On Locating Objects by Their Distinguishing Features in Multi-Sensory Images," Computer Graphics and Image Processing, December 1973.

Tenenbaum, J. and S. Weyl, "A Region-Analysis Subsystem for Interactive Scene Analysis," Tech Note 104, Stanford Research International, Menlo Park CA, June 1975.

Tenenbaum, J., and H. Barrow, "Experiments in Interpretation Guided Segmentation," Tech Note 123, Stanford Research Institute International, March 1976.

Thompson, C., "Depth Perception in Stereo Computer Vision," SAIL Memo AIM-268, Computer Science Dept, Stanford University, Stanford CA, October 1975.

Thompson, W., "Combining Motion and Contrast for Segmentation," Tech Report 79-7, Institute of Technology, University of Minnesota, Minneapolis MN, March 1979.

Tsotsos, J., "A Prototype Motion Understanding System," Tech Report 93, Department of Computer Science, University of Toronto, Canada, June 1976.

Ullman, S., "The Correspondence Process in Motion Perception," Proc. ARPA Workshop on Image Understanding, May 1978.

Waltz, D., "Generating Semantic Descriptions from Drawings of Scenes with Shadows," Tech Report TR-271, AI lab MIT, Cambridge MA, 1972.

AD-A107 565 MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INF--ETC F/6 9/2  
COMPUTER INTERPRETATION OF A DYNAMIC IMAGE FROM A MOVING VEHICL--ETC(U)  
MAY 81 T D WILLIAMS N00014-75-C-0459  
UNCLASSIFIED COINS-TR-81-22 NL

MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INF--ETC F/6 9/2  
COMPUTER INTERPRETATION OF A DYNAMIC IMAGE FROM A MOVING VEHICLE--ETC(U)  
MAY 81 T D WILLIAMS N00014-75-C-0459  
COINS-TR-81-22 NL

UNCLASSIFIED

 $\Delta O_F = 4$ 

200

END

DATE \_\_\_\_\_

FILMED

8-11

DTIC

Williams, T. and J. Lowrance, "Model Building in the VISIONS High Level System," Tech Report 77-1, Computer and Information Science Dept, University of Massachusetts, Amherst MA, January 1977.

Williams, T., et. al., "Model-Building in the VISIONS System," Proc. 5th International Joint Conference on Artificial Intelligence, MIT, Cambridge MA, August 1977.

Woods, W., "Whats in a Link?" from Bobrow and Collins (Ed.) Representation and Understanding, Academic Press, 1975.

Yakimovsky, Y., and J. Feldman, "A Semantics-Based Decision Theory Region Analyzer", Proc. 3rd International Conference on Artificial Intelligence, Stanford CA, August 1973.

York, B., "A Primer on Polynomial Interpolation and Splines," Tech Report 79-5, Computer and Information Science Department, University of Massachusetts, Amherst MA, 1979.

York, B., "Shape Representation and Computer Vision," Ph.D. thesis in preperation, Computer and Information Science Department, University of Massachusetts, Amherst, 1981.

## A P P E N D I X I

### RELAXATION AND THE UPDATE RULE

The objective of applying relaxation procedures to obtain a segmentation is to use the context immediately surrounding a central pixel to update that pixel's label. The label updating is a parallel iterative process, with a new set of labels replacing the current set of labels on each iteration. The effect of applying this update rule is to reduce noise points, and to smooth regions which are jagged, or remove those which are only one or two pixels wide.

A number of techniques exist for performing this relaxation, and a number of update rules have been explored (see Nagin 1979). The simplest is a "discrete" rule, using only a single label at each pixel in a three by three (nine pixel) neighborhood immediately surrounding the central pixel. The new label chosen is the mode, or most frequent label in the neighborhood. In case of ties, either the label at the central pixel is chosen if it is among the contenders, or one of the contenders is chosen randomly.

One simple option with this update rule is to count the central pixel more than once. Then, if there is some supporting evidence for the label (through a small number of similarly labeled pixels in the neighborhood), there is a greater likelihood that the pixel's label will remain unchanged. In our processing we chose to count the central pixel three times. Thus, the total number of counts in the nine pixel neighborhood would be 11, and in cases where only two distinct labels are present, there would only need to be three that are the same as the central label to leave it unchanged. The same rule, counting the central pixel three times, was used for initial segmentation and for segmentation of the error image.

## A P P E N D I X I I

### THE UVW COLOR SPACE

The original color data are recorded by imaging the original film through red, green and blue filters onto a digitizing vidicon. These three filters produce a tri-stimulus data set which can reproduce almost all the colors recorded on the original medium. By considering these values as a three-dimensional vector, one can manipulate the data to transform it into any other color description. Hue, saturation and intensity are common terms for one such transform, Y, I and Q are the three used for the color television standard. For a clear and complete description of color space transforms please refer to Pratt 1978. The UVW color space has been used by colormetric investigators who are interested in converting between various color spaces, and has been used very effectively for color image segmentation by (Nagin 1979).

The V and W color features are used in this thesis to form the two-dimensional histogram for feature generation. The V color dimension is basically a red versus green opponent measure, and the W is a white versus black measure. The computation for V and W are as



follows:

$$V = -.354 r - .797 g + .905 b$$

$$W = .605 r + .801 g + .392 b$$

In this notation  $r$ ,  $g$  and  $b$  are the red, green and blue values respectively. The  $V$  and  $W$  values that result are scaled into the range of  $[0,63]$  in order to generate the two-dimensional histograms used in the feature generation process.

## A P P E N D I X I I I

## COLOR AND

## TEXTURE FEATURES USED FOR OBJECT INTERPRETATION

Eleven features were computed for use in the formation of prototypes for each object, and subsequent measurements for object interpretation. They fall into two categories. The first is point features, where only the values (red, green and blue) at each point contribute to the average and standard deviation feature values. The second category is texture features, where a neighborhood of intensity values around the central point contribute to the feature value.

The point features are:

raw red	=	r
raw green	=	g
raw blue	=	b
Y	=	.299 r + .587 g + .144 b
I	=	.596 r + .274 g + .322 b
Q	=	.211 r + .523 g + .312 b
Saturation	=	1 - min(nr, ng, nb)
Intensity	=	(r + g + b) / 3

The notation  $r$ ,  $g$  and  $b$  are the red, green and blue values, and  $nr$ ,  $ng$  and  $nb$  are the normalized red, green and blue values respectively. The normalized values are computed as follows:

$$\begin{aligned} nr &= \frac{r}{r + g + b} \\ ng &= \frac{g}{r + g + b} \\ nb &= \frac{b}{r + g + b} \end{aligned}$$

The first texture feature is a contrast measure computed over a small area. First the intensity difference between the central pixel and each of the four adjacent pixels is computed. These values are then squared and averaged. This gives a measure of the average square of intensity difference surrounding a pixel and therefore is a measure of the strength of texture.

The last two features are edge contrast per-unit area, at each of two orientations, with non-maximal edges suppressed. The vertical measure is computed by first forming the signed intensity difference between each horizontal pair of pixels, thereby encoding vertical edge

contrast as the magnitude, and the direction of contrast as the sign. All those vertical edges which have an adjacent parallel edge with the same sign and a greater magnitude are considered non-maximal edges and are suppressed to zero. The average of the absolute value of the remaining edges is the vertical texture measure desired. The horizontal edge contrast with non-maxima suppression is computed in a similar manner. These two measures are intended to roughly capture orientation dependent characteristics of texture.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER COINS-TR-81-22	2. GOVT ACCESSION NO. AD-A107565	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  COMPUTER INTERPRETATION OF A DYNAMIC IMAGE FROM A MOVING VEHICLE		5. TYPE OF REPORT & PERIOD COVERED INTERIM
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)  Thomas D. Williams		8. CONTRACT OR GRANT NUMBER(s)  ONR N00014-75-C-0459
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer and Information Science Department University of Massachusetts Amherst, Massachusetts 01003		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS  Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE May 1981
		13. NUMBER OF PAGES 295
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)  UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Distribution of this document is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) region analysis motion analysis image processing scene analysis time varying imagery		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  The goal of this report is the design and implementation of a computer program that constructs an interpretation of images of a natural scene, in particular one imaged while the camera is in a moving automobile. The succession of images is to be interpreted in terms of surfaces and objects in three-dimensional space.		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

The agreement between image dynamics and an internal surface model of the environment is measured by comparing a pair of temporally disparate images (two movie frames). Using the model, an image taken at one location can be transformed into a synthetic image of the scene as it would be viewed from another location. This synthesis accounts for point displacements and occlusion effects as predicted by the internal model. Differences between the real and the synthetic images are then used as an error measure in a search that refines the model. Once the model is refined, unresolved errors are used to correct the initial surface model by resegmenting the image into a better approximation of the surfaces in the environment.

This surface model refinement is followed by an object identification phase. Size and color attributes measured from the derived internal model are compared with stored attributes for objects. The result is the identification of some of the scene objects.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)